James D. Veltrop
Jeremy C. Lowe
AXINN, VELTROP & HARKRIDER LLP
90 State House Square
Hartford, CT 06103
Tel: (860) 275-8100

James P. Doyle
AXINN, VELTROP & HARKRIDER LLP
114 West 47th Street
New York, NY 10036
Tel: (212) 728-2200

Attorneys for Plaintiff
PRODUCTION RESOURCE GROUP, L.L.C.

08 CIV. 6331

JUDGE KARAS

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

PRODUCTION RESOURCE GROUP, L.L.C., Plaintiff, v. SGM TECHNOLOGY FOR LIGHTING SpA and TECHNI-LUX, INC., Defendants.	Civil Action No. ________ **COMPLAINT AND JURY DEMAND**

Plaintiff Production Resource Group, L.L.C. ("Plaintiff" or "PRG"), by its undersigned attorneys, for its Complaint herein against Defendants SGM Technology for Lighting SpA and Techni-Lux, Inc. (collectively "Defendants"), alleges upon knowledge with respect to its own acts, and upon information and belief as to other matters, as follows:

THE PARTIES

1. Plaintiff PRG is a Delaware limited liability company having its principal place of business at 539 Temple Hill Road, New Windsor, New York 12553. PRG is in the business of, among other things, designing, manufacturing and selling sophisticated lighting fixtures, controllers and associated products for use in entertainment and display environments.

2. Defendant SGM Technology for Lighting SpA is an Italian company having its principal place of business at Via Pio La Torre, 1, 61010 Tavullia PU, Italy.

3. Defendant Techni-Lux, Inc. is a Florida corporation having its principal place of business as 10779 Satellite Boulevard, Orlando, Florida 32837. Techni-Lux, Inc. is a United States distributor of products sold by SGM Technology for Lighting SpA.

NATURE OF THE ACTION

4. This action arises under the patent laws of the United States, 35 U.S.C. § 1, *et seq.*

JURISDICTION AND VENUE

5. This Court has jurisdiction over the claims asserted herein pursuant to 28 U.S.C. §§ 1331 and 1338(a).

6. This Court has personal jurisdiction over Defendants because Defendants regularly transact business within this judicial district and have committed acts of patent infringement within this judicial district.

7. Venue is proper in this Court pursuant to 28 U.S.C. §§ 1391(b) and (c) and 1400(b).

COUNT ONE
INFRINGEMENT OF UNITED STATES PATENT NO. 6,711,411

8. PRG incorporates each of the preceding paragraphs of this Complaint as if fully set forth herein.

9. PRG exclusively owns United States Patent No. 6,711,411 ("the '411 patent"), entitled "Programmable Light Beam Shape Altering Device Using Programmable Micromirrors," issued on August 3, 2004. See Exhibit A.

10. In violation of 35 U.S.C. § 271, Defendants have made, used, offered for sale or sold within the United States or imported into the United States products that infringe one or more claims of the '411 patent, including the Giotto Digital 1500 products.

11. Defendants' infringement has been with knowledge of the '411 patent and has been willful.

12. PRG has suffered and will continue to suffer damages and irreparable injuries unless Defendants' infringement of the '411 patent is enjoined.

COUNT TWO
INFRINGEMENT OF UNITED STATES PATENT NO. 6,891,656

13. PRG incorporates each of the preceding paragraphs of this Complaint as if fully set forth herein.

14. PRG exclusively owns United States Patent No. 6,891,656 ("the '656 patent"), entitled "Pixel Based Gobo Record Control Format," issued on May 10, 2005. See Exhibit B.

15. In violation of 35 U.S.C. § 271, Defendants have made, used, offered for sale or sold within the United States or imported into the United States products that infringe one or more claims of the '656 patent, including the Giotto Digital 1500 products.

16. Defendants' infringement has been with knowledge of the '656 patent and has been willful.

17. PRG has suffered and will continue to suffer damages and irreparable injuries unless Defendants' infringement of the '656 patent is enjoined.

COUNT THREE
INFRINGEMENT OF UNITED STATES PATENT NO. 6,919,916

18. PRG incorporates each of the preceding paragraphs of this Complaint as if fully set forth herein.

19. PRG exclusively owns United States Patent No. 6,919,916 ("the '916 patent"), entitled "Method and Device for Creating a Facsimile of an Image," issued on July 19, 2005. See Exhibit C.

20. In violation of 35 U.S.C. § 271, Defendants have made, used, offered for sale or sold within the United States or imported into the United States products that infringe one or more claims of the '916 patent, including the Giotto Digital 1500 products.

21. Defendants' infringement has been with knowledge of the '916 patent and has been willful.

22. PRG has suffered and will continue to suffer damages and irreparable injuries unless Defendants' infringement of the '916 patent is enjoined.

COUNT FOUR
INFRINGEMENT OF UNITED STATES PATENT NO. 7,161,562

23. PRG incorporates each of the preceding paragraphs of this Complaint as if fully set forth herein.

24. PRG exclusively owns United States Patent No. 7,161,562 ("the '562 patent"), entitled "Multilayer Control of Gobo Shape," issued on January 9, 2007. See Exhibit D.

25. In violation of 35 U.S.C. § 271, Defendants have made, used, offered for sale or sold within the United States or imported into the United States products that infringe one or more claims of the '562 patent, including the Giotto Digital 1500 products.

26. Defendants' infringement has been with knowledge of the '562 patent and has been willful.

27. PRG has suffered and will continue to suffer damages and irreparable injuries unless Defendants' infringement of the '562 patent is enjoined.

PRAYER FOR RELIEF

WHEREFORE, PRG respectfully requests that this Court enter judgment in its favor and against Defendants and grant the following relief:

A. A judgment that Defendants have infringed the '411, '656, '916 and '562 patents directly, or by inducement or contribution, in violation of 35 U.S.C. § 271;

B. A judgment that Defendants' infringement of the '411, '656, '916 and '562 patents has been willful;

C. An order, pursuant to 35 U.S.C. § 283, enjoining Defendants and all persons in active concert or participation with Defendants from any further infringement of the '411, '656, '916 and '562 patents;

D. An order, pursuant to 35 U.S.C. § 284, awarding PRG damages adequate to compensate for Defendants' infringement of the '411, '656, '916 and '562 patents, in amounts to be determined at trial, but in no event less than reasonable royalties;

E. An order, pursuant to 35 U.S.C. § 284, trebling all damages awarded to PRG in view of Defendants' willful infringement of the '411, '656, '916 and '562 patents;

F. An order, pursuant to 28 U.S.C. § 1961 and 35 U.S.C. § 284, awarding to PRG interest on the damages and its costs incurred from this action;

G. An order, pursuant to 35 U.S.C. § 285, awarding to PRG its attorneys' fees;

H. An order directing Defendants to recall from distribution and destroy their entire stock of infringing products within the United States; and

I. Such other and further relief as the Court may deem just and proper.

JURY DEMAND

In accordance with Fed. R. Civ. P. 38 and 39, Plaintiff PRG asserts its right under the Seventh Amendment to the United States Constitution and demands a trial by jury on all issues that may be so tried.

Dated: July 15, 2008

Respectfully submitted,

James D. Veltrop
Jeremy C. Lowe
AXINN, VELTROP & HARKRIDER LLP
90 State House Square
Hartford, CT 06103
Tel: (860) 275-8100
Fax: (860) 275-8101

James P. Doyle
AXINN, VELTROP & HARKRIDER LLP
114 West 47th Street
New York, NY 10036
Tel: (212) 728-2200
Fax: (212) 728-2201

Attorneys for Plaintiff
PRODUCTION RESOURCE GROUP, L.L.C.

EXHIBIT A

US006711411B1

(12) **United States Patent**
Ruffini

(10) **Patent No.: US 6,711,411 B1**
(45) **Date of Patent: Mar. 23, 2004**

(54) **MANAGEMENT OF SYNCHRONIZATION NETWORK**

(75) Inventor: **Stefano Ruffini**, Rome (IT)

(73) Assignee: **Telefonaktiebolaget LM Ericsson (publ)**, Stockholm (SE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 433 days.

(21) Appl. No.: **09/706,727**

(22) Filed: **Nov. 7, 2000**

(51) **Int. Cl.**[7] **H04B 15/00**

(52) **U.S. Cl.** **455/502**; 455/428; 370/350; 370/355

(58) **Field of Search** 455/502, 402, 455/418, 428–439, 26.1, 78, 560, 191.3, 445; 370/351, 355, 356, 357, 358, 360

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,939,726 A	*	7/1990	Flammer et al.	370/400
4,987,536 A		1/1991	Humblet	
5,086,428 A		2/1992	Perlman et al.	
5,128,926 A		7/1992	Perlman et al.	
5,265,092 A		11/1993	Soloway et al.	
5,398,317 A		3/1995	Nugent	
5,453,977 A	*	9/1995	Flammer et al.	370/254
5,687,168 A		11/1997	Iwata	
5,805,593 A		9/1998	Busche	
5,825,772 A		10/1998	Dobbins et al.	
5,875,179 A		2/1999	Tikalsky	
5,881,243 A		3/1999	Zaumen et al.	
5,920,566 A		7/1999	Hendel et al.	
5,964,841 A		10/1999	Rekhter	
5,987,521 A		11/1999	Arrowood et al.	
6,097,718 A		8/2000	Bion	
6,198,728 B1	*	3/2001	Hulyalkar et al.	370/310.1
6,393,118 B1	*	5/2002	Bhusri	379/219
2002/0193101 A1	*	12/2002	McAlinden	455/419

FOREIGN PATENT DOCUMENTS

WO	WO98/15077	4/1998
WO	WO98/37651	8/1998

OTHER PUBLICATIONS

"Transmission and Multiplexing (TM); Synchronization Network Engineering," European Telecommunications Standard Institute (ETSI), ETSI EG 201 793 v1.1.1 (2000–10), pp. 1–86.

"Transmission and Multiplexing (TM); Generic Requirements of Transport Functionality of Equipment; Part 6–1: Sychronization Layer Functions, " European Telecommunication Standards Institute (ETSI), ETS 300 417–6–1, Aug. 1998, pp. 1–28.

"OSPF Version 2, " J. Moy, Ascend Communications, Inc., Apr. 1998, pp. 1–244.

"Definitions and Terminology for Synchronization Networks", ITU–T Recommendation G.810, Aug. 1996, pp. 1–20.

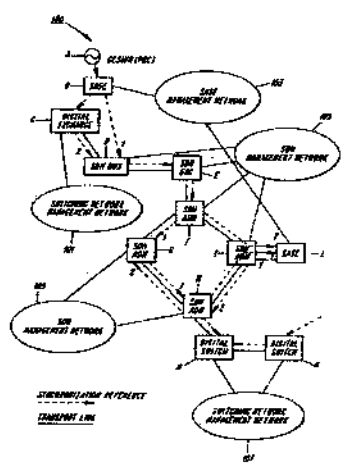
* cited by examiner

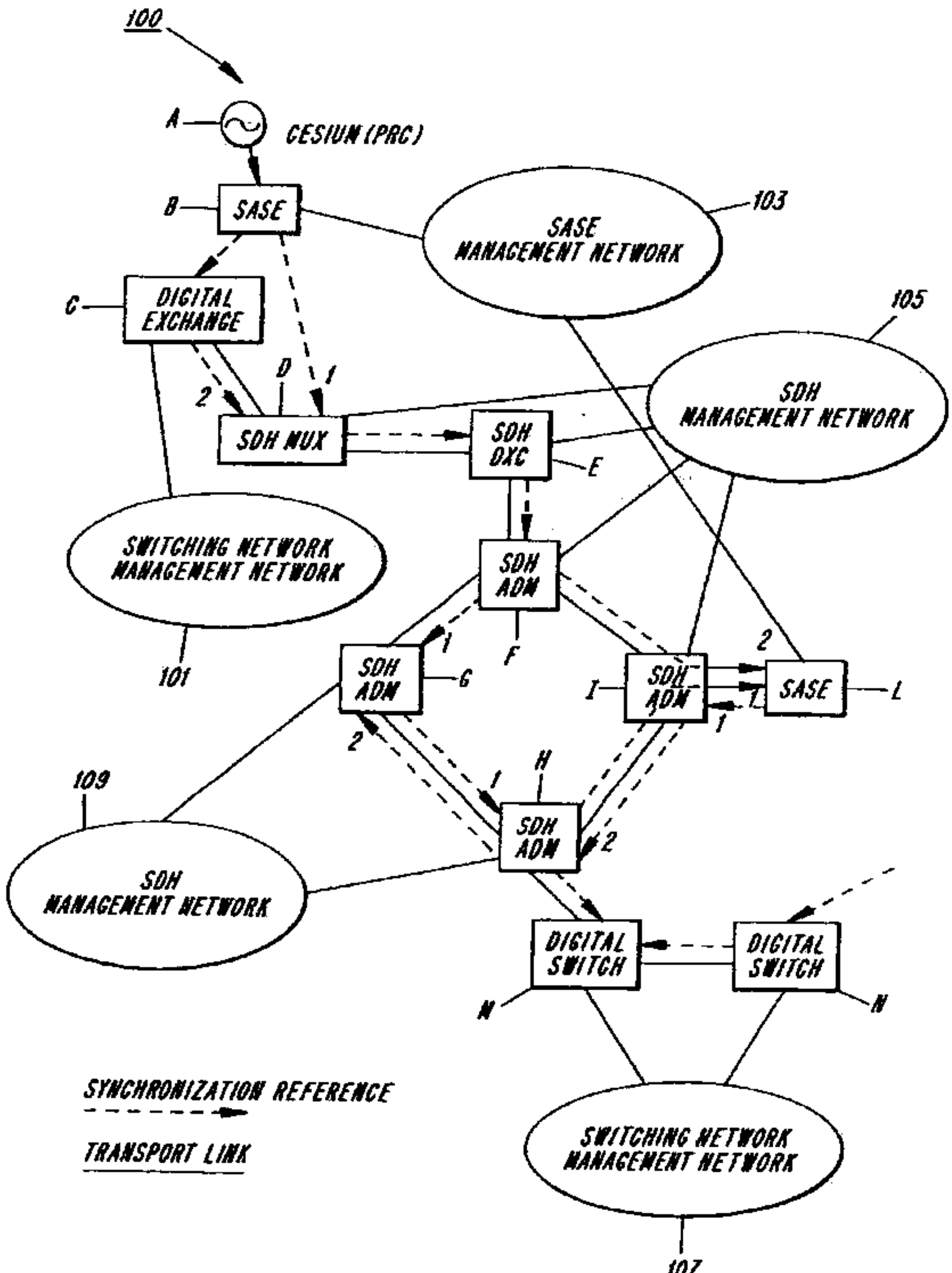
Primary Examiner—Congvan Tran

(57) **ABSTRACT**

Operation of a synchronization network that includes a number of nodes and reference clock distribution logic includes, at each of the nodes, storing a table that represents a most recent status of the synchronization network. Reference clocks are then distributed to each node in the synchronization network. If a change in synchronization status occurs at a node in the synchronization network, this change is detected. The table at the node is updated to represent an updated status of the synchronization network. Then, a synchronization network management protocol is used to distribute the updated status to other nodes in the synchronization network. Other nodes may change their status in response to receipt of the updated information, and information about these additional updates are distributed throughout the synchronization network in a similar manner, thereby facilitating control of the synchronization network.

24 Claims, 11 Drawing Sheets





100
A
CESIUM (PRC)
B
SASE
103
SASE
MANAGEMENT NETWORK
C
DIGITAL
EXCHANGE
105
D
1
2
SDH MUX
SDH
DXC
E
SDH
MANAGEMENT NETWORK
SWITCHING NETWORK
MANAGEMENT NETWORK
SDH
ADM
101
1
F
2
SDH
ADM
G
I
SDH
ADM
SASE
L
1
1
2
H
109
1
SDH
ADM
2
SDH
MANAGEMENT NETWORK
DIGITAL
SWITCH
DIGITAL
SWITCH
M
N
SYNCHRONIZATION REFERENCE
TRANSPORT LINK
SWITCHING NETWORK
MANAGEMENT NETWORK
107

FIG. 1

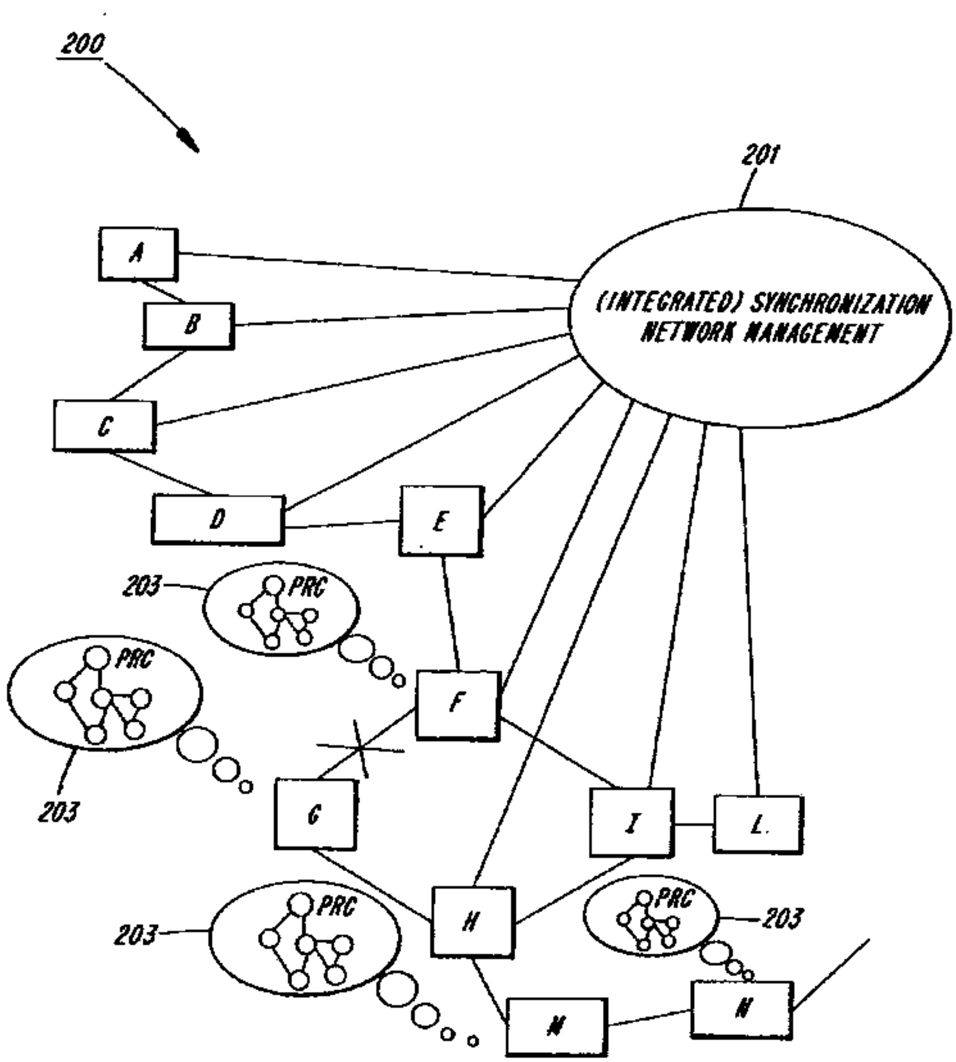

200
201
(INTEGRATED) SYNCHRONIZATION NETWORK MANAGEMENT
A
B
C
D
E
F
G
H
I
L
M
N
PRC
203

FIG. 2

300

NETWORK ELEMENT (301)	SYNCHRONIZATION FROM (303)	PRIORITY (305)	TRACEABLE TO (307)	STATUS (309)	SUPERVISION RESULT (MTIE, TDEV, FDEV, SSM, ETC.) (311)
A	–	–	–	NETWORK PRC (G.811)	
B	A	1	A	LOCKED (G.811)	MTIE=....
B	B (INTERNAL CLOCK)	2	B	STAND-BY (G.812)	MTIE=....
C	B	1	A-B	LOCKED (G.811)	MTIE=....
C	B	2	A-B	STAND-BY (G.811)	MTIE=....
C	C (INTERNAL CLOCK)	3	C	STAND-BY (G.812)	MTIE=....
D	B	1	A-B	LOCKED (G.811)	MTIE=....
D	C	2	A-B-C	STAND-BY (G.811)	MTIE=....
D	D (INTERNAL CLOCK)	3	D	STAND-BY (G.813)	MTIE=....
E	D	1	A-B-D	LOCKED (G.811)	MTIE=....
E	E (INTERNAL CLOCK)	2	E	STAND-BY (G.812)	MTIE=....
F	E	1	A-B-D-E	LOCKED (G.811)	MTIE=....
F	F (INTERNAL CLOCK)	2	F	STAND-BY (G.813)	MTIE=....
G	F	1	A-B-D-E-F	LOCKED (G.811)	MTIE=....
G	H	2	A-B-D-E-F-G-H	STAND-BY (G.811)	MTIE=....
G	G (INTERNAL CLOCK)	3	G	STAND-BY (G.813)	MTIE=....

FIG. 3A

NETWORK ELEMENT (301)	SYNCHRONIZATION FROM (303)	PRIORITY (305)	TRACEABLE TO (307)	STATUS (309)	SUPERVISION RESULT (MTIE, TDEV, FDEV, SSM, ETC.) (311)
H	G	1	A-B-D-E-F-G	LOCKED (G.811)	MTIE=...
H	I	2	A-B-D-E-F-G-H-L-I	STAND-BY (G.811)	MTIE=...
H	H (INTERNAL CLOCK)	3	H	STAND-BY (G.813)	MTIE=...
I	L	1	A-B-D-E-F-G-H-(I)-L	LOCKED (G.811)	MTIE=...
I	I(INTERNAL CLOCK)	2	I	STAND-BY (G.813)	MTIE=...
L	I (NON LOCKED, FROM H)	1	A-B-D-E-F-G-H-(I)	LOCKED (G.811)	MTIE=...
L	I (NON LOCKED, FROM F)	2	A-B-D-E-F-(I)	STAND-BY (G.811)	MTIE=...
L	L (INTERNAL CLOCK)	3	L	STAND-BY (G.812)	MTIE=...
...	...	...	...	...	...

FIG. 3B

300'

301 NETWORK ELEMENT	303 SYNCHRONIZATION FROM	305 PRIORITY	307 TRACEABLE TO	309 STATUS	311 SUPERVISION RESULT (MTIE, TDEV, FDEV, SSM, ETC.)
A	–	–	–	NETWORK PRC (G.811)	
B	A	1	A	LOCKED (G.811)	MTIE = ...
B	B (INTERNAL CLOCK)	2	B	STAND-BY (G.812)	MTIE = ...
C	B	1	A-B	LOCKED (G.811)	MTIE = ...
C	B	2	A-B	STAND-BY (G.811)	MTIE = ...
C	C (INTERNAL CLOCK)	3	C	STAND-BY (G.812)	MTIE = ...
D	B	1	A-B	LOCKED (G.811)	MTIE = ...
D	C	2	A-B-C	STAND-BY (G.811)	MTIE = ...
D	D (INTERNAL CLOCK)	3	D	STAND-BY (G.813)	MTIE = ...
E	D	1	A-B-D	LOCKED (G.811)	MTIE = ...
E	E (INTERNAL CLOCK)	2	E	STAND-BY (G.812)	MTIE = ...
F	E	1	A-B-D-E	LOCKED (G.811)	MTIE = ...
F	F (INTERNAL CLOCK)	2	F	STAND-BY (G.813)	MTIE = ...
G	F	1	–	LOCKED (G.811) -> FAILURE	MTIE = ... ALARM
G	H	2	A-B-D-E-F-G-H	STAND-BY (G.811)	MTIE = ...
G	G (INTERNAL CLOCK)	3	G	STAND-BY -> LOCKED (G.813)	MTIE = ...

FIG. 4A

NETWORK ELEMENT (301)	SYNCHRONIZATION FROM (303)	PRIORITY (305)	TRACEABLE TO (307)	STATUS (309)	SUPERVISION RESULT (MTIE, TDEV, FDEV, SSM, ETC.) (311)
H	G	1	A-B-D-E-F-G.	LOCKED (G.811->G.813)	MTIE=....
H	I	2	A-B-D-E-F-G-H-(I)-L-I	STAND-BY (G.811)	MTIE=....
H	H (INTERNAL CLOCK)	3	H	STAND-BY (G.813)	MTIE=....
I	L	1	A-B-D-E-F-(I)-L	LOCKED (G.811)	MTIE=....
I	I (INTERNAL CLOCK)	2	I	STAND-BY (G.813)	MTIE=....
L	I (NON LOCKED, FROM H)	1	A-B-D-E-F-G-H-(I)	LOCKED-> STAND-BY (G.811->G.813)	MTIE= ALARM....
L	I (NON LOCKED, FROM F)	2	A-B-D-E-F-(I)	STAND-BY-> LOCKED (G.811)	MTIE=....
L	L (INTERNAL CLOCK)	3	L	STAND-BY (G.812)	MTIE=....
...	...	...	...	...	...

FIG. 4B

300"

301 303 305 307 309 311

NETWORK ELEMENT	SYNCHRONIZATION FROM	PRIORITY	TRACEABLE TO	STATUS	SUPERVISION RESULT (MTIE, TDEV, FDEV, SSM, ETC.)
A	–	–	–	NETWORK PRC (G.811)	
B	A	1	A	LOCKED (G.811)	MTIE = ...
B	B (INTERNAL CLOCK)	2	B	STAND-BY (G.812)	MTIE = ...
C	B	1	A-B	LOCKED (G.811)	MTIE = ...
C	B	2	A-B	STAND-BY (G.811)	MTIE = ...
C	C (INTERNAL CLOCK)	3	C	STAND-BY (G.812)	MTIE = ...
D	B	1	A-B	LOCKED (G.811)	MTIE = ...
D	C	2	A-B-C	STAND-BY (G.811)	MTIE = ...
D	D (INTERNAL CLOCK)	3	D	STAND-BY (G.813)	MTIE = ...
E	D	1	A-B-D	LOCKED (G.811)	MTIE = ...
E	E (INTERNAL CLOCK)	2	E	STAND-BY (G.812)	MTIE = ...
F	E	1	A-B-D-E	LOCKED (G.811)	MTIE = ...
F	F (INTERVAL CLOCK)	2	F	STAND-BY (G.813)	MTIE = ...
G	F	1	–	FAILURE	MTIE = ... ALARM
G	H	2	A-B-D-E-F-(I)-L-I-H	STAND-BY (G.811) -> LOCKED (G.811)	MTIE = ...
G	G (INTERNAL CLOCK)	3	G	LOCKED -> STAND-BY (G.813)	MTIE = ...

FIG. 5A

301 NETWORK ELEMENT	303 SYNCHRONIZATION FROM	305 PRIORITY	307 TRACEABLE TO	309 STATUS	311 SUPERVISION RESULT (MTIE, TDEV, FDEV, SSM, ETC.)
H	G	1	G	STAND-BY (G.813)	MTIE = ...
H	I	2	A-B-D-E-F-(I)-L-I	STAND-BY-> LOCKED (G.811)	MTIE = ...
H	H (INTERNAL CLOCK)	3	H	STAND-BY (G.813)	MTIE = ...
I	L	1	A-B-D-E-F-(I)-L	LOCKED (G.811)	MTIE ....
I	I (INTERNAL CLOCK)	2	I	STAND-BY (G.813)	MTIE = ...
L	I (NON LOCKED, FROM H)	1	A-B-D-E-F-(I)-L-I-H-(I)	STAND-BY (G.811)	MTIE = ...
L	I (NON LOCKED, FROM F)	2	A-B-D-E-F-(I)	LOCKED (G.811)	MTIE = ...
L	L (INTERNAL CLOCK)	3	L	STAND-BY (G.812)	MTIE = ...
...	...	...	...	...	...

FIG. 5B

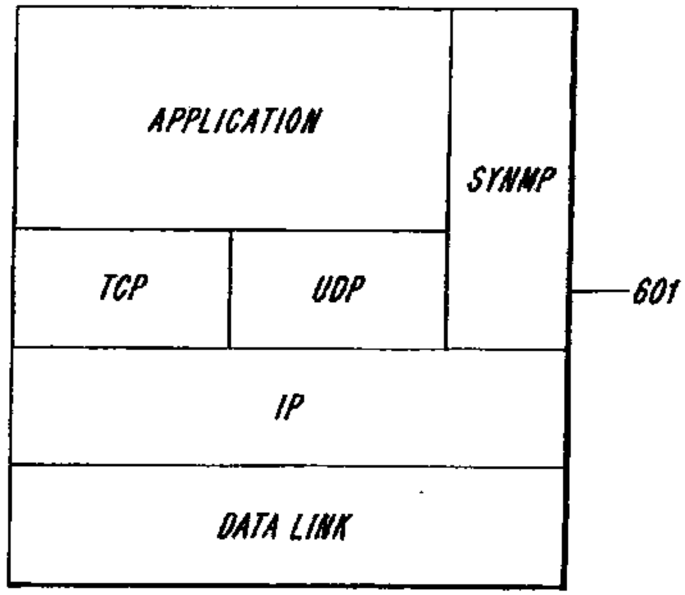

APPLICATION
SYNMP
TCP
UDP
601
IP
DATA LINK

FIG. 6

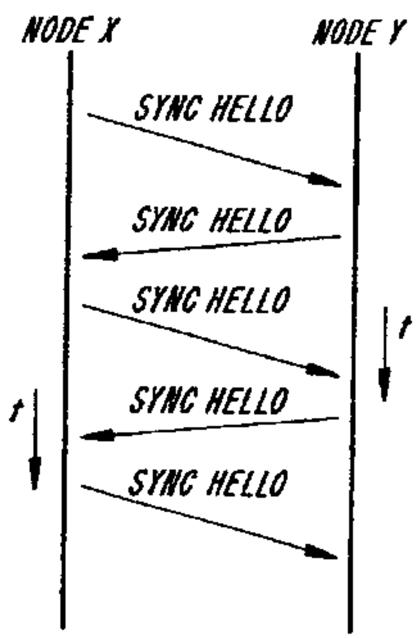

NODE X
NODE Y
SYNC HELLO
SYNC HELLO
SYNC HELLO
t
SYNC HELLO
t
SYNC HELLO

FIG. 7

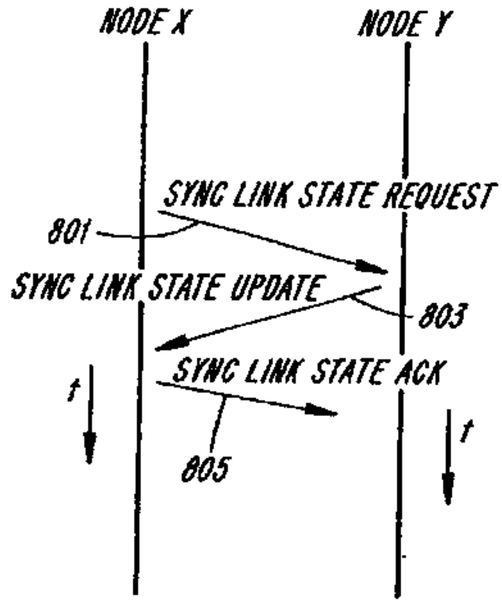

NODE X
NODE Y
SYNC LINK STATE REQUEST
801
SYNC LINK STATE UPDATE
803
SYNC LINK STATE ACK
805
t
t

FIG. 8

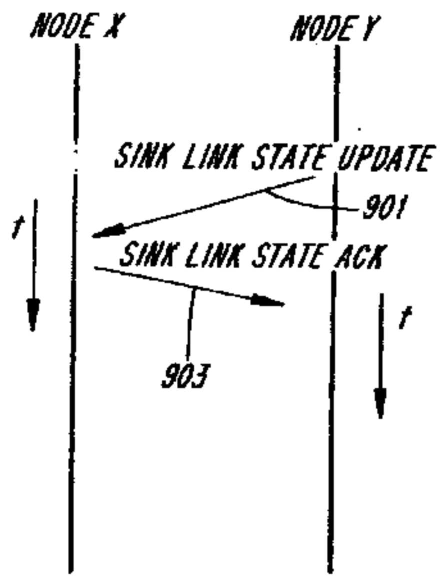

NODE X
NODE Y
SINK LINK STATE UPDATE
901
SINK LINK STATE ACK
903
t
t

FIG. 9

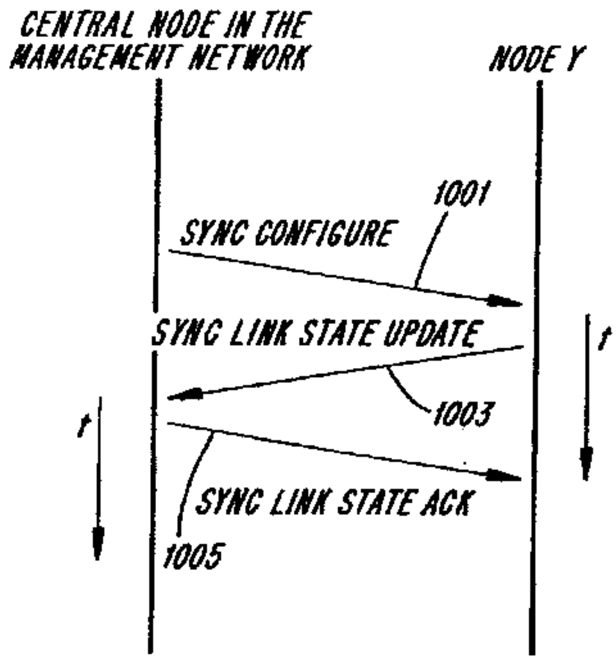

CENTRAL NODE IN THE MANAGEMENT NETWORK
NODE Y
1001
SYNC CONFIGURE
SYNC LINK STATE UPDATE
t
1003
t
SYNC LINK STATE ACK
1005

FIG. 10

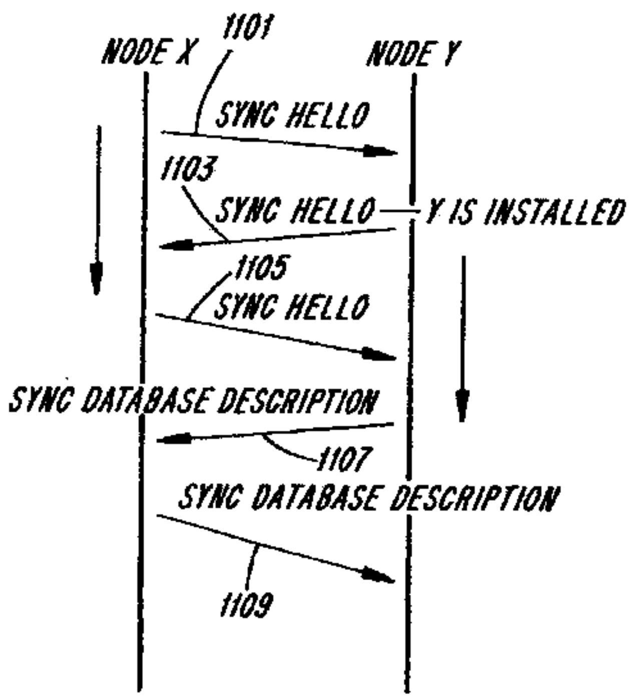

1101
NODE X
NODE Y
SYNC HELLO
1103
SYNC HELLO
Y IS INSTALLED
1105
SYNC HELLO
SYNC DATABASE DESCRIPTION
1107
SYNC DATABASE DESCRIPTION
1109

FIG. 11

MANAGEMENT OF SYNCHRONIZATION NETWORK

BACKGROUND

The invention relates to synchronization in telecommunication systems.

Digital communication networks often require a common timing reference to operate accurately. That is, the clocks in one node of the network should operate at the same speed as the clocks in other nodes of the network. Failure to provide synchronized clocks will lead to Jitter and Wander, which in turn can lead to such problems as transmission errors and buffer under/overflow. A network cannot maintain low error rates under such conditions, and ultimately may require some degree of unavailability to rectify the situation.

To provide for a common timing reference, digital communication networks include synchronization networks, whose job it is to ensure that a common timing reference is used throughout the network. One such synchronization network is described in European Telecommunication Standards Institute (ETSI) document European Guide (EG) 201 793 v1.1.1 (2000-10), entitled "Transmission and Multiplexing (TM); Synchronization Network Engineering", which is hereby incorporated herein by reference in its entirety. This document describes the various elements that make up a synchronization network, and the principles of operation by which such a network distributes accurate timing information from so-called Primary Reference Clocks (PRCs) to the clocks located in other pieces of equipment throughout the network. PRCs are the highest quality clocks in a network, and are usually based on a free-running Caesium Beam oscillator giving a very accurate clock performance.

FIG. 1 is a block diagram of an exemplary digital communication network **100** that includes a synchronization network. For purposes of illustration, the exemplary network **100** is a telecommunications network, and therefore includes, at network nodes, equipment that is well-known in the art. In the figure, transport links are indicated by solid lines, and synchronization reference links are shown by dashed lines that include an arrow at one end to indicate the source and recipient of the reference clock signal. Where a node has the possibility of receiving a reference clock from more than one source, primary reference links (i.e., those synchronization links that are preferred to be used for supplying a reference clock from one node to another) are denoted by the number "1" next to the dashed line indicating the link. Secondary reference links (i.e., those synchronization links that are used when the primary synchronization link is unavailable) are denoted by the number "2" next to the dashed line indicating the link.

The exemplary network **100** utilizes the Synchronous Digital Hierarchy (SDH), which is a standard technology for synchronous data transmission on optical media. It is the international equivalent of the Synchronous Optical Network (SONET). To facilitate the following discussion, the various nodes of the network are given reference characters A, B, C, D, E, F, G, H, I, L, M, and N.

In a fully synchronized network, all sources should be ultimately traceable to a PRC. In the exemplary network, this is the PRC A. The PRC A supplies its high quality clocking signal ("clock") to Stand Alone Synchronization Equipment (SASE) B. A SASE is a piece of synchronization equipment that contains a Synchronization Supply Unit (SSU), which is a high quality slave clock. The SASE B distributes its clock signal to a Digital Exchange C (which, in alternative embodiments, could be a Telephone Exchange) and also to an SDH multiplexer (MUX) D.

The SDH MUX D distributes its clock signal to an SDH Digital Cross Connect unit (SDH DXC) E, which in turn distributes its clock signal to an SDH Add Drop Multiplexer (ADM) F. The clock supplied by the SDH ADM F is then supplied to each of two more SDH ADMs G and I. The reference link between the SDH ADM F and the SDH ADM G is a primary link.

Rather than using the supplied clock signal itself, the SDH ADM I operates in a "bypass" mode (commonly named "NON-SETS locked", where "SETS" stands for "Synchronous Equipment Timing Source"), whereby the synchronization clock is merely forwarded directly to the SASE L. This is common when, for example, the ADM and SASE are implemented in the same building. Essentially, the SASE L is the real recipient of the synchronization clock supplied by SDH ADM F, and this clock is treated as a secondary link. In the exemplary embodiment, the SASE L's primary link is supplied (through the SDH ADM I operating in "bypass" mode) by an SDH ADM H.

Despite its bypass function, the SDH ADM I does require a synchronization clock, and in the exemplary embodiment this is supplied by the SASE L.

The SDH ADM I supplies its synchronization clock to the SDH ADM H, and this is treated as a secondary link. The SDH ADM H's primary link is supplied by the SDH ADM G. To permit reconfigurability, the SDH ADM H is also coupled to supply a synchronization clock to the SDH ADM G, and this is treated as a secondary link by the SDH ADM G.

In accordance with the exemplary embodiment, the SDH ADM H also supplies a synchronization clock to a digital switch M, which also receives a synchronization clock from the digital switch N. The remainder of the exemplary network is not shown, since this is not important to understanding the invention.

It is very important that the synchronization network be planned in such a way so as to avoid the occurrence of timing loops, both during normal operation as well as when a malfunction prevents one or more nodes from supplying their reference clocks to their planned recipient nodes. A timing loop is created when a clock is directly or indirectly synchronized to itself. In a timing loop situation, all the clocks belonging to the loop can show a large frequency offset compared to nominal frequency and are likely to be isolated from the rest of the synchronization network. To avoid timing loops, elements in a ring should be provided with means that enable the possible generation of timing loops to be discovered. Such elements are usually connected such that they each have at least two synchronization sources, so that when one source is discovered to cause a timing loop, there is at least the possibility of avoiding it by selecting one of the alternative sources. For example, suppose that the reference link between nodes F and G is cut. In this situation, the SDH ADM G will respond by looking to node H to supply the necessary reference clock. However, under normal circumstances, node H expects to receive its reference clock from node G. It is apparent that a timing loop will occur here unless node H also responds to the break between nodes F and G by looking to another source for its reference clock. It is important that the clock supplied by this alternative source also not ultimately be derived from the clock at node G or from the clock at node H in order to avoid a timing loop.

In SDH networks, the use of Synchronous Status Messages (SSMs) provides some help with avoiding timing

loops. An SSM is a signal that is passed over a synchronization interface to indicate the Quality-Level of the clock that the interface is ultimately traceable to; that is, the grade-of-clock to which it is synchronized directly or indirectly via a chain of network element clocks (the "synchronization trail"), however long this chain of clocks is. In a fully synchronized network, all sources should ultimately be traceable to a PRC, and there is a predefined code to indicate this. Another code, "Do Not Use for Synchronization", is used to prevent timing loops and is transmitted in the opposite direction on interfaces used to synchronize an equipment's clock.

Although the SSM algorithm is a good concept in some applications like SDH or SONET rings, it is unable to guarantee that all timing loops will be prevented, because it only provides information about the quality of the traceable synchronization reference source, but not information about the actual physical source. See, for example, chapter 4.13 of ETS 300 417-6-1, "Generic requirements of transport functionality of equipment: Synchronization layer function", which is hereby incorporated herein by reference in its entirety. Another drawback of the SSM algorithm is that it is often not supported by SASE or by Network Elements other than SDH/SONET Network elements (i.e., it can only be used between SDH/SONET Network Elements).

It is noted that timing loops can cause severe disturbances in the traffic network, but that the effect of these disturbances ver seldom gives a readily discernable indication of where the failure in the synchronization network occurred. It is therefore important to provide effective ways for managing synchronization networks so that, when failures occur in the network, it can be determined how to rearrange the network to maintain an acceptable quality of synchronization without creating timing loops.

Today, the management of synchronization networks is distributed among several platforms. The reason for this is that a synchronization network very often consists of different types of equipment that can be either dedicated to synchronization (e.g., SASE), or to both synchronization and traffic (e.g., and SDH multiplexer or a digital switch). As a result, several management systems (e.g., one for the SASE network, one for the SDH equipment, one for the switching network, etc.) have to be maintained in parallel. This situation is illustrated in FIG. 1, in which a first Switching Network Management Network **101** manages the Digital Exchange at node C; a SASE Management Network **103** manages the SASE at nodes B and L; a first SDH Management Network **105** manages the SDH Mux at node D, the SDH DXC at node E, and the SDH ADMs at nodes F and I; a second Switching Network Management Network **107** manages the Digital Switches at nodes M and N; and a second SDH Management Network **109** manages the SDH ADMs at nodes G and H. This situation is not unrealistic, since the "same" type of equipment (e.g., an SDH ADM) may be manufactured by different vendors who design their equipment using incompatible equipment management strategies. Such equipment may, nonetheless, be connected together in a single network to achieve diverse goals, such as supporting mobile network functions on the one hand, and hard-wired telephone functions on the other.

This problem of distributed management will only get worse in the future as new types of equipment (e.g., Internet Protocol (IP) routers) increase their synchronization function (due to new network application of this equipment), which will in turn require that this new equipment be managed from a synchronization perspective.

The above describes the best case situation of the conventional synchronization management approach. In

practice, the network management of synchronization networks more often does not exist, or is incomplete. Because of this, conventional synchronization networks not only have to be well planned, but also need to be continuously maintained by maintenance personnel who may need to be physically present at the numerous sites. In most cases, this physical presence at each site is impossible due to geographical distribution of the synchronization network.

For the above reasons, it is very difficult for an operator of a conventional system to have full control and visibility of the synchronization network. Better synchronization management techniques and systems are therefore desired.

SUMMARY

It should be emphasized that the terms "comprises" and "comprising", when used in this specification, are taken to specify the presence of stated features, integers, steps or components; but the use of these terms does not preclude the presence or addition of one or more other features, integers, steps, components or groups thereof.

In accordance with one aspect of the present invention, the foregoing and other objects are achieved in methods and apparatuses that operate a synchronization network that includes a plurality of nodes and logic that distributes reference clocks to each of the nodes. Operation of the synchronization network includes, at each of the nodes, storing a table that represents a most recent status of the synchronization network. A change in synchronization status at a first node in the synchronization network is detected, and at the first node, the table is updated to represent a first updated status of the synchronization network. A synchronization network management protocol is then used to distribute the first updated status to other nodes in the synchronization network. In this way, each node in the synchronization network can have complete information about the most recent status of the synchronization network, thereby facilitating management of the network.

In another aspect of the invention, one or more other nodes may change their own status in response to receipt of information identifying a change in status at another node. By similarly updating their own tables and using the synchronization network management protocol to distribute the updated status to other nodes in the synchronization network, an iterative process is used that results in each node in the synchronization network having the most recent status about the synchronization network.

In some embodiments, the synchronization network may include an integrated synchronization network management node that also receives the synchronization network status information, and uses this to control all of the nodes in the synchronization network.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects and advantages of the invention will be understood by reading the following detailed description in conjunction with the drawings in which:

FIG. **1** is a block diagram of an exemplary digital communication network **100** that includes a synchronization network;

FIG. **2** is a block diagram of an exemplary synchronization network having a centralized control mechanism in accordance with the invention;

FIGS. *3a* and *3b* together make up an exemplary table that, in accordance with the invention, corresponds to the Synchronization Network **200**;

FIGS. 4a and 4b together make up an exemplary updated table 300' that, in accordance with the invention, corresponds to the Synchronization Network 200 after a malfunction has occurred between nodes F and G;

FIGS. 5a and 5b together make up an exemplary table 300" that, in accordance with the invention, corresponds to the Synchronization Network 200 after reconfigurations have been made to accommodate the malfunction that occurred between nodes F and G;

FIG. 6 is a diagram of a protocol stack that illustrates the relationship between SYNMP and other protocol layers including IP, in accordance with the invention;

FIG. 7 is a timing diagram that illustrates the periodic exchange of "Hello" packets between two nodes X and Y, in accordance with an aspect of the invention;

FIG. 8 is a timing diagram that illustrates a situation in which a node X requests, and subsequently receives, up-to-date synchronization status information about another node Y in the synchronization network;

FIG. 9 is a timing diagram that illustrates a situation in which a node Y detects a change in its own status, and informs another node X of this information;

FIG. 10 is a timing diagram that illustrates how a central node in the management network uses the new SYNMP protocol to configure another node (node Y) in the synchronization network;

FIG. 11 is a timing diagram that illustrates how two nodes can synchronize their databases by means of the inventive SYNMP protocol; and

DETAILED DESCRIPTION

The various features of the invention will now be described with respect to the figures, in which like parts are identified with the same reference characters.

These and other aspects of the invention will now be described in greater detail in connection with a number of exemplary embodiments. To facilitate an understanding of the invention, many aspects of the invention are described in terms of sequences of actions to be performed by elements of a computer system. It will be recognized that in each of the embodiments, the various actions could be performed by specialized circuits (e.g., discrete logic gates interconnected to perform a specialized function), by program instructions being executed by one or more processors, or by a combination of both. Moreover, the invention can additionally be considered to be embodied entirely within any form of computer readable carrier, such as solid-state memory, magnetic disk, optical disk or carrier wave (such as radio frequency, audio frequency or optical frequency carrier waves) containing an appropriate set of computer instructions that would cause a processor to carry out the techniques described herein. Thus, the various aspects of the invention may be embodied in many different forms, and all such forms are contemplated to be within the scope of the invention. For each of the various aspects of the invention, any such form of embodiment may be referred to herein as "logic configured to" perform a described action, or alternatively as "logic that" performs a described action.

Considered at a high level, the invention involves the use of node addresses and a suitable protocol for providing the equipment at each node of the synchronization network with a complete description of the dynamic status of the synchronization network at any point in the network. This same information can then be made available in a central management center. In this way, decisions regarding any necessary network rearrangement in case of failures can be made autonomously by the equipment at any node, or alternatively can be made by the equipment at a central management center.

An exemplary embodiment of the invention will now be described in greater detail using principles that are known in connection with link state routing protocols such as the Open Shortest Path First (OSPF) data routing protocol. These principles can be advantageously applied in the present invention relating to synchronization network management because they provide for fast reaction in case of changes in network topology, small overhead traffic, and flexibility in rearranging the network based on proper optimization algorithms.

This is not to be understood to mean that the conventional link state routing protocols are useful for managing synchronization networks—they are not since, for example, they do not provide for the exchange of data pertinent to the synchronization network management functions (e.g., data pertaining to synchronization trails and status). However, the OSPF routing protocol enables all routers in an IP network to have a complete picture of the IP network for purposes of being able to route each data packet from its source node to its intended destination node, and it will now be described how, in accordance with the invention, these concepts are usefully applied in a new way to facilitate the management of synchronization networks.

In OSPF, providing all routers with a complete picture of the IP network is accomplished by having each router send updates of its link state (i.e., information about changes in the status of connected links) to other neighboring routers in the IP network. In a recursive way, the same information is spread throughout the network, until each router has the information, and can create a complete picture of the IP network that can be used in the routing of IP packets.

In accordance with the invention, a similar approach is used in Synchronization Networks to provide each node in the Synchronization Network with complete information about the link status of all of the other nodes in the Synchronization Network. To do this, the information needed is:

- The active synchronization reference and its status (that is, the identity of the traceable synchronization source and its quality; and
- The identities of the stand-by (back-up) synchronization references and their status.

FIG. 2 is a block diagram of an exemplary synchronization network 200 having a centralized control mechanism in accordance with the invention. To facilitate a comparison between the invention and conventional synchronization network management techniques, the synchronization network 200 comprises nodes, denoted A, B, C, D, E, F, G, H, I, L, M, and N, that are nominally linked together in the same configuration as that of the synchronization network 100 depicted in FIG. 1. The synchronization network 200 differs from a conventional network, however, in that it further includes, at each node, the ability to route synchronization link state information throughout the synchronization network 200, so that each node in the synchronization network 200 maintains complete information about the dynamic link status of the synchronization network 200 at any point in the network. In this exemplary embodiment, the synchronization network 200 further includes an Integrated Synchronization Network Management Network 201 that is capable of managing all of the nodes in the synchronization network. Provision of an Integrated Synchronization Network Man-

agement Network 201 is not essential to the invention, since as explained above, this function can alternatively be distributed among the various nodes, which each have complete information about the dynamic link status of the synchronization network 200.

In one aspect of the invention, each node in the Synchronization Management Network 200 is assigned an address (e.g., an IP address) that uniquely identifies that node. For purposes of convenience, the reference characters A, B, C, D, E, F, G, H, I, L, M, and N, that are used herein to denote the nodes, will also be used herein to represent the respective addresses of those nodes.

In the exemplary embodiment, the equipment at each node has an interface that is capable of supporting IP, and all such equipment is connected to an IP network. It is emphasized, however, that the use of IP standards is not essential to the invention, and that other telecommunication protocol standards could be substituted therefor, so long as those other protocols permit the communication of information between nodes as described below.

In another aspect of the invention, each node further has a storage device that stores a table that defines the relationship between each node and other physical links in the Synchronization Network 200. In FIG. 2, this is represented schematically by the several configuration information diagrams 203.

FIGS. 3*a* and 3*b* together make up an exemplary table 300 that corresponds to the Synchronization Network 200. In a first column 301 of the table 300, each of the nodes in the Synchronization Network 200 is defined. For each of these nodes, an entry in a second column 303 of the table 300 identifies the node's source of synchronization. For example, node A is a PRC (refer to FIG. 1), and thus has no other source. Node B has a link to receive synchronization from node A, and this is indicated in the second column 303. Moreover, for node B this is the preferred source of synchronization, so in a third column 305 of the table 300, there is an indication that this source is to be given highest priority (e.g., priority "1").

Many nodes have more than one possible source of synchronization. For example, node B has a link to receive synchronization from node A, as indicated above, but also has an internal clock that it can alternatively use as a synchronization reference. Each of these is assigned a relative priority that indicates an order of preference of use of these possible synchronization sources. So, for example, node B's highest priority synchronization reference is received from node A, and its next highest priority synchronization reference (in this case, priority "2") comes from node B's own internal clock.

A fourth column 307 in the table 300 indicates the succession of nodes through which the synchronization reference can be traced. For example, node A is a PRC, so there are no nodes through which its synchronization reference is derived. For node B, the synchronization reference is traceable to node A when the highest priority reference (i.e., the one provided by node A) is used, and alternatively traceable to B itself when B's own internal clock is used.

A fifth column 309 indicates, for the network element, the status of the corresponding synchronization reference. Possible states include: "Network PRC (G.811)"; "locked (G.811)"; "stand-by (G.812)"; and "stand-by (G.813)", the latter being a lower quality clock than that defined by G.812. These states are well-known in the art, and are defined, for example, in ITU-T recommendation G.811 (2/97), "Timing characteristics of Primary Reference Clocks"; and ITU-T recommendation G.812 (2/97), "Timing characteristics of Slave Clocks"; and ITU-T recommendation G.813 (8/96), "Timing characteristics of SDH Equipment Slave-Clocks (SEC)", all of which are hereby incorporated herein by reference in their entireties.

Finally, a sixth column 311 in the table 300 indicates a Supervision Result, such as: Maximum Time Interval Error (MTIE), Time Deviation (TDEV), Frequency Deviation (FDEV), Synchronization Status Message (SSM), and the like. These Supervision Results are the result of usually periodical supervision, and are set by the routine in the network element performing the supervision tests. These results may be read by the operator to check the quality of a synchronization network, but may also be used by the synchronization network in an automatic manner in order to reconfigure the synchronization network (e.g., in case a link exhibits poor quality). The Supervision Results MTIE and TDEV are well known in the art, and are defined in ITU-T recommendation G.810 (5/96), "Definitions and terminology for synchronization networks", which is hereby incorporated herein by reference in its entirety. The Supervision Result SSM is also well known and is defined in the above-referenced ETSI document EG 201 783. Frequency Deviation is also a well-known concept, and need not be defined here.

In order to avoid excessively large tables, the synchronization network can optionally be divided up into parts, with each part containing only data for a related subset of the entire synchronization network.

The discussion will now focus on how the contents of the table are modified in the event of a failure in the synchronization network. Suppose that a malfunction (e.g., a break in a cable) occurs that renders the synchronization link between nodes F and G inoperative. This means that node G will have to obtain synchronization from another source, and in order to avoid timing loops, other nodes may need to be reconfigured as well. In accordance with an aspect of the invention, this effort is coordinated by utilizing the new protocol (described in greater detail below) to distribute table updates, so that each node (including the one associated with the Integrated Synchronization Network Management network 201) will have complete information upon which to base reconfiguration decisions. The information changes piece-wise, so a number of updated tables are created and distributed in succession.

For example, after detecting the loss of its synchronization source, node G would examine its own table and discover that it cannot immediately select to receive its synchronization reference from node H, since this would cause a timing loop. (The table shows that the node H's source is traceable to node G.) So, node G opts instead to receive its synchronization reference from its own internal clock.

Node G updates the table 300 to reflect this change, and distributes it to the other nodes in the synchronization network. As the table is distributed from node to node, changes are made in a piecemeal fashion to adjust to the previous changes. FIGS. 4A and 4B illustrate an intermediate state of the table 300' after it has been updated to reflect several changes. In particular, a Supervision Result indicating "ALARM" has been generated in Node G as a result of the supervisor detecting something not working, such as Loss of Frame, or MTIE exceeding a certain threshold. This "ALARM" in turn causes Node G's highest priority link to transition from a "locked" state to a "FAILURE" state (the "->" sign in the table indicates a transition from one state to another). In response to this failure, Node G's third priority synchronization reference source is now shown as transi-

tioning from a "standby" mode to a "locked" state (meaning that it is being used by node G). This highlights the fact that in the reference selection algorithm, the condition of avoiding timing loops is given greater importance than choosing the reference having the highest priority. At this stage of the reconfiguration, node H is still receiving its synchronization reference from node G, but its table entries change to show a transition in quality from a G.811 quality reference to a G.813 quality reference. Another intermediate change reflected in the table **300'** is exemplified by the entry for Node L, which also shows a Supervision Result of "ALARM" because the reference clock from node G is in the traceback path of Node L's primary synchronization source. In response, Node L has reconfigured itself to select the reference clock from its secondary source, Node I (operating in bypass mode to pass along the reference clock from node F). This is shown in the table **300'** as the priority 1 source transitioning from a "locked" to a "standby" mode, and the priority 2 source transitioning from a "standby" to "locked" mode.

Some time later, the table **300"** will look as shown in FIGS. 5A and 5B: At this point, node H has chosen to receive its synchronization reference from its second highest priority link, since this provides a better quality than the synchronization reference being supplied by node G. This is shown in the table **300"** as a transitioning from a "standby" state to a "locked" mode of operation. Node G, in turn, chooses to receive the synchronization reference coming from node H, since this is a higher quality reference than its own internal clock, and because this can now be done without the risk of creating timing loops: This is shown in the table **300"** as node G's priority 2 reference source transitioning from "standby" to "locked" mode, while the priority 3 reference source transitions from a "locked" mode to a "standby" mode.

Note that node G's synchronization reference from node H is no longer derived from node G. Note also that, for Node L, the highest priority reference clock source is no longer traceable through G, but is traceable to Node L itself. This could create a timing loop if Node L were to select this as the source. However, as shown in the table **300"**, Node L is receiving its reference clock from its secondary source, which does not suffer from this problem.

The table **300"** shown in FIGS. 5A and 5B is not the final one in this iterative updating process, since the changes shown in this table will in turn cause entries for other nodes to change. For example, after node G has completed its transition from use of its internal clock (priority 3) to use of the reference clock from node H, the "traceable to" entry for node H, priority 1, should be updated to show a chain of nodes that includes node "H". To avoid cluttering the description with unnecessary details, these and other subsequent changes are not shown or described here further.

It is noted that each node can now make its own reconfiguration decisions automatically without any reliance on the conventional SSM algorithm, simply by relying on its own database which now reflects the most recent state of the synchronization network.

In another aspect of the invention, changes in synchronization status at a node cause the updated table to be propagated throughout the network. This may happen for a while without any changes in synchronization source being changed by any of the nodes. (Alternatively, nodes can be designed to respond immediately to information that indicates that one of the nodes in the succession of nodes through which the synchronization reference can be traced has experienced a failure.) Ultimately, the table with the changed status information makes its way to the Integrated Synchronization Network Management Network **201** (assuming that one is provided in the embodiment). The Integrated Synchronization Network Management can then take appropriate steps to work around the synchronization failure.

In yet another aspect of the invention, to route the table **300** throughout the synchronization network, a new protocol, herein referred to as Synchronization Network Management Protocol (SYNMP), is used.

The SYNMP protocol may have a similar approach as that used in the Link State routing protocol (e.g., OSPF, which is a well-known protocol for distributing routing information to all routers in a network, and for deciding the best routing to use). FIG. 6 is a diagram of a protocol stack that illustrates the relationship between the SYNMP protocol **601** and other protocol layers including IP. Like OSPF, the new SYNMP protocol **601** would be defined to operate directly over IP. However, unlike OSPF, instead of routing data, the new SYNMP protocol **601** transfers information that makes it possible for each node of the synchronization network (as well as at a centralized Synchronization Network Management center **201**) to get information regarding synchronization trails and status. With this information, the network element is able to find the best alternative synchronization sources without the risk of creating timing loops and in accordance with the synchronization network plan.

Of course, dynamic modifications to the synchronization network plan can be made by personnel operating at, for example, the centralized Synchronization Network Management center **201**. However, in alternative embodiments of the invention, the operation of the synchronization network can be made completely automated. For example, each network element might choose the most convenient reference based on an algorithm that satisfies one or more criteria, such as minimizing the length of trails, or providing good reliability (e.g., based on geographical diversity).

The following types of packets are defined in an exemplary embodiment of the SYNMP protocol **601**:

"Sync Hello" packet: This packet periodically informs the connected nodes of the sender's synchronization status. In practice, the time interval between transmissions of this packet can be relatively high (e.g., on the order of hours) because the stability of network elements in a telecommunication network is relatively high (relative to network elements in datacom networks).

"Sync Database Description" packet: This packet may be used in an initializing phase to send information about active and stand-by synchronization references and their quality status (e.g., as a result of MTIE/TDEV supervision). A "Sync Database Description" packet is sent back as acknowledgment.

"Sync Link State request": This packet is used when synchronization data is to be updated, such as after the "Sync Database Description" exchange, in case part of the database is found to be out of date.

"Sync Link State Update": This packet may be sent to inform other nodes of changes in the synchronization references state (e.g., after failures). It may be sent autonomously by the node when it detects the changes in its own status. Alternatively, the "Sync Link State Update" packet may be sent by the node in response to receipt of a "Sync Link State request" packet from another node.

"Sync Link state Acknowledgment": This packet may be used to acknowledge receipt of a "Sync Link State Update" packet.

In addition to the above packet types, the protocol should provide a mechanism for communicating specific commands that determine remote configuration and operation of the synchronization function of the equipment. These commands would be used by an operator at a remote centralized location to manage the entire synchronization network. Exemplary commands include, but are not limited to:

a command to set which reference should be accepted as a synchronization reference;

a command to set the priorities for the different synchronization references;

one or more commands to set the alarm thresholds for the supervision parameters.

FIGS. 7–11 are timing diagrams showing examples how a number of the above-described packets are exchanged in accordance with the SYNMP protocol defined by the invention. FIG. 7 is a timing diagram that illustrates the periodic exchange of "Sync Hello" packets between two nodes X and Y. As explained above, each "Sync Hello" packet may include information about the synchronization status of the sending node.

FIG. 8 is a timing diagram that illustrates a situation in which a node X requests, and subsequently receives, up-to-date synchronization status information about another node Y in the synchronization network. To accomplish this, node X first sends a Sync Link State Request packet 801 to node Y. In response, node Y sends the requested information to node X by means of a Sync Link State Update packet 803. To complete this transaction, Node X then acknowledges receipt of the Sync Link State Update packet 803 by sending a Sync Link State Acknowledge packet 805 to node Y.

FIG. 9 is a timing diagram that illustrates a situation in which a node Y detects a change in its own status, and informs another node X of this information. In particular, node Y communicates the up-to-date synchronization status information to node X by means of a Sync Link State Update packet 901. To complete this transaction, Node X then acknowledges receipt of the Sync Link State Update packet 901 by sending a Sync Link State Acknowledge packet 903 to node Y.

FIG. 10 is a timing diagram that illustrates how a central node in the management network uses the new SYNMP protocol to configure another node (node Y) in the synchronization network. First, a packet that includes suitable synchronization configuration commands 1001 is sent from the central node to the node Y. Node Y carries out the commanded reconfiguration (not shown). This reconfiguration results in a synchronization status change at node Y, which is reflected as an update to node Y's table 300. Accordingly, node Y communicates its most up-to-date synchronization status information to the central node by means of a Sync Link State Update packet 1003. To complete this transaction, the central node then acknowledges receipt of the Sync Link State Update packet 1003 by sending a Sync Link State Acknowledge packet 1005 to node Y.

FIG. 11 is a timing diagram that illustrates how two nodes can synchronize their databases by means of the inventive SYNMP protocol. In the example, a node X sends out a Sync Hello packet 1101 as part of its normal functioning. Sometime later, a different node, node Y, is installed. Node Y sends out its own Sync Hello packet 1103 as part of its normal functioning. Node X receives node Y's Sync Hello packet, and now sends out its own Sync Hello packet 1105 to node Y. In response to receipt of node X's Sync Hello packet 1105, node Y sends a Sync Database Description packet 1107 to node X. The Sync Database Description packet 1107 includes information about node Y's active and stand-by synchronization references and their quality status. Node X uses this information to update its own table 300. Finally, Node X acknowledges receipt of node Y's Sync Database Description packet 1107 by sending a Sync Database Description packet 1109 back to node Y.

FIG. 12 is a flowchart illustrating, at a high level, steps that are performed in operating the synchronization network in accordance with the invention. First, the tables in each node are initialized (using the new SYNMP protocol 601) to reflect the initial state of the synchronization network (step 1201). Each node (including the Integrated Synchronization Network Management node, if one is included in the embodiment) now has a complete picture of how reference clocks are propagated throughout the network.

Next, the synchronization network is operated, using well-known techniques, to distribute synchronization reference clocks to each node (step 1203). So long as there are no problems or other changes ("NO" path out of decision block 1205) this continues indefinitely. However, whenever a change in status occurs at a node ("YES" path out of decision block 1205), the table in the affected node is updated (step 1207). The new SYNMP protocol 601 is then used to propagate this information to all other nodes in the synchronization network, including the Integrated Synchronization Network Management node, if one is included in the embodiment (step 1209). The Sync Link State Update packet can be used for this task.

As updated information is propagated throughout the synchronization network, one or more nodes may respond with their own changes in status/configuration, therefore requiring further updates to the tables 300. Thus, through well-known iterative techniques, the new SYNMP protocol 601 is used to propagate the further updated information to all other nodes in the synchronization network, including the Integrated Synchronization Network Management node, if one is included in the embodiment (step 1211).

Eventually, no further changes are made at any nodes, and each node again has a complete picture of how reference clocks are propagated throughout the network. The synchronization network can then again be operated, using well-known techniques, to distribute synchronization reference clocks to each node (step 1203).

The above described techniques make it possible to completely integrate all of the synchronization network management functions into a single (centralized) node. This is especially important now that new types of equipment (IP routers in particular) are starting to need to be controlled by a Synchronization Network Management Network as well.

The above described techniques provide an additional advantage in that they make it possible to completely automate the planning and dynamic modification of the synchronization network. For example, a suitable algorithm for automatically planning a synchronization network might be based on such considerations as shortest paths, quality or reliability. Automatic decisions are advantageous because they can be made in a very short period of time, especially considering that, in accordance with the invention, any change in the synchronization network will be very quickly made visible at any point in the network.

The invention has been described with reference to a particular embodiment. However, it will be readily apparent to those skilled in the art that it is possible to embody the invention in specific forms other than those of the preferred embodiment described above. This may be done without departing from the spirit of the invention.

For example, an exemplary embodiment of the Synchronization Network Management Protocol has described that

includes a number of exemplary packet and command types that can be exchanged between nodes in a synchronization network to provide each node in the synchronization network with most recent information describing the synchronization routing through the network. It should be understood, however, that the exemplary packet and command types are not essential to the invention, but rather are presented merely to illustrate one possible embodiment of the protocol. Alternative embodiments, employing other packet types and/or command types, are also considered to be within the scope of the invention, so long as those other packet types and/or command types are designed for use in distributing synchronization-related information throughout a synchronization network.

Thus, the preferred embodiment is merely illustrative and should not be considered restrictive in any way. The scope of the invention is given by the appended claims, rather than the preceding description, and all variations and equivalents which fall within the range of the claims are intended to be embraced therein.

What is claimed is:

1. A method of operating a synchronization network that includes a plurality of nodes, wherein reference clocks are distributed to each node in the synchronization network, the method comprising:

at each of the nodes, storing a table that includes a most recent status of the synchronization network, said network synchronization status including, for all of the nodes in the network, information indicating a primary synchronization reference for each node, whether each node has one or more secondary synchronization references, a priority indication for each of the synchronization references for each node, succession of nodes through which each synchronization reference can be traced from each node to the reference, and an indication of which reference is currently being used by each node;

detecting a change in synchronization status at a first node in the synchronization network;

at the first node, updating the table to represent a first updated status of the synchronization network; and

using a synchronization network management protocol to distribute the first updated status from the first node to other nodes in the synchronization network.

2. The method of claim 1, further comprising:

at a second node in the synchronization network, changing the synchronization information for the second node in response to receiving the first updated status generated by the first node;

at the second node, updating the table to include a second updated status of the synchronization network; and

using the synchronization network management protocol to distribute the second updated status from the second node to other nodes in the synchronization network.

3. The method of claim 1, further comprising:

receiving the first updated status in an integrated synchronization network management network node;

generating by the integrated synchronization network management network node, one or more synchronization network control commands in response to the first updated status; and

using the synchronization network management protocol to distribute the one or more synchronization network control commands from the integrated synchronization network management network node to one or more targeted nodes in the synchronization network.

4. The method of claim 1 wherein the table includes a supervision result for each node in the network, said supervision result indicating a quality level of at least one synchronization parameter.

5. A method of maintaining synchronization configuration status information in a synchronization network that includes a first node and second node, the method comprising:

periodically sending a first "Sync Hello" packet from the first node to the second node, wherein the first "Sync Hello" packet includes most recent information about synchronization status of the first node;

periodically sending a second "Sync Hello" packet from the second node to the first node, wherein the second "Sync Hello" packet includes most recent information about synchronization status of the second node;

detecting in the first node, a failure of a synchronization link between the first node and a primary synchronization source for the first node; and

sending a "Sync Link State Update" packet from the first node to the second node, wherein the "Sync Link State Update" packet includes information about the failure of the synchronization link between the first node and the primary synchronization source for the first node.

6. The method of claim 5, further comprising:

in response to receipt of the "Sync Link State Update" packet in the second node, sending a "Sync Link State Acknowledgment" packet from the second node to the first node.

7. The method of claim 5, further comprising:

initializing a third node in the synchronization network, said initializing step including sending a "Sync Link State Request" packet from the third node to the first node to update the synchronization configuration status information for the synchronization network.

8. The method of claim 7, further comprising:

in response to receipt of the "Sync Link State Request" packet from the third node, sending a "Sync Link State Update" packet from the first node to the third node, wherein the "Sync Link State Update" packet includes information about the synchronization status of the first node.

9. The method of claim 8, further comprising:

in response to receipt of the "Sync Link State Update" packet in the third node, sending a "Sync Link State Acknowledgment" packet from the third node to the first node.

10. A method of configuring a first node in a synchronization network, the method comprising:

sending a configuration packet from a central management network node to the first node, wherein the configuration packet includes one or more synchronization configuration commands;

in the first node, carrying out the one or more synchronization configuration commands, and generating therefrom an updated synchronization status for the first node;

sending a "Sync Link State Update" packet from the first node to the central management network node, wherein the "Sync Link State Update" packet includes the updated synchronization status for the first node; and

in response to receipt of the "Sync Link State Update" packet in the central management network node, sending a "Sync Link State Acknowledge" packet from the central management network node to the first node

acknowledging receipt of the updated synchronization status for the first node.

11. A method of updating a database in a first node in a synchronization network, the method comprising:

populating the database in the first node with synchronization information for the first node;

sending a "Sync Hello" packet from the first node to a second node in the synchronization network, said "Sync Hello" packet including information about the synchronization status of the first node;

in response to receipt of the "Sync Hello" packet in the second node:

updating a database in the second node with the information about the synchronization status of the first node; and

sending a first "Sync Database Description" packet from the second node to the first node, wherein the first "Sync Database Description" packet includes information about the synchronization status of the second node; and

in the first node, updating the database with the information about the synchronization status of the second node.

12. The method of claim 11, further comprising:

acknowledging receipt of the first "Sync Database Description" packet in the first node by sending a second "Sync Database Description" packet from the first node to the second node.

13. An apparatus for operating a synchronization network that comprises a plurality of nodes and logic that distributes reference clocks to each node in the synchronization network, the apparatus comprising:

at each of the nodes, logic that stores a table that includes a most recent status of the synchronization network, said network synchronization status including, for all of the nodes in the network, information indicating a primary synchronization reference for each node, whether each node has one or more secondary synchronization references, a priority indication for each of the synchronization references for each node, a succession of nodes through which each synchronization reference can be traced from each node to the reference, and an indication of which reference is currently being used by each node;

logic that detects a change in synchronization status at a first node in the synchronization network;

at the first node, logic that updates the table to represent a first updated status of the synchronization network; and

logic that uses a synchronization network management protocol to distribute the first updated status from the first node to other nodes in the synchronization network.

14. The apparatus of claim 13, further comprising:

at a second node in the synchronization network, logic that changes the synchronization information for the second node in response to receiving the first updated status generated by the first node;

at the second node, logic that updates the table to include a second updated status of the synchronization network; and

logic that uses the synchronization network management protocol to distribute the second updated status from the second node to other nodes in the synchronization network.

15. The apparatus of claim 13, further comprising:

logic that receives the first updated status in an integrated synchronization network management network node;

logic in the integrated synchronization network management network node that generates one or more synchronization network control commands in response to the first updated status; and

logic that uses the synchronization network management protocol to distribute the one or more synchronization network control commands from the integrated synchronization network management network node to one or more targeted nodes in the synchronization network.

16. The apparatus of claim 13, wherein the table includes a supervision result for each node in the network, said supervision result indicating a quality level of at least one synchronization parameter.

17. An apparatus for maintaining synchronization configuration status information in a synchronization network that includes a first node and a second node, the apparatus comprising:

logic that periodically sends a first "Sync Hello" packet from the first node to the second node, wherein the first "Sync Hello" packet includes most recent information about synchronization status of the first node;

logic that periodically sends a second "Sync Hello" packet from the second node to the first node, wherein the second "Sync Hello" packet includes most recent information about synchronization status of the second node;

logic in the first node that detects a failure of a synchronization link between the first node and a primary synchronization source for the first node; and

logic that sends a "Sync Link State Update" Packet from the first node to the second node, wherein the "Sync Link State Update" Packet includes information about the failure of the synchronization link between the first node and the primary synchronization source for the first node.

18. The apparatus of claim 17, further comprising:

logic in the second node that sends a "Sync Link State Acknowledgment" packet from the second node to the first node in response to receipt of the "Sync Link State Update" packet by the second node.

19. The apparatus of claim 17, further comprising:

logic in a third node in the synchronization network that initializes the third node, including sending a "Sync Link State Request" packet from the third node to the first node to update the synchronization configuration status information for the synchronization network.

20. The apparatus of claim 19, further comprising:

logic in the first node that sends a "Sync Link State Update" packet from the first node to the third node in response to receipt of the "Sync Link State Request" packet from the third node, wherein the "Sync Link State Update" packet includes information about the synchronization status of the first node.

21. The apparatus of claim 20, further comprising:

logic in the third node that sends a "Sync Link State Acknowledgment" packet from the third node to the first node in response to receipt of the "Sync Link State Update" packet in the third node.

22. An apparatus for configuring a first node in a synchronization network, the apparatus comprising:

logic in a central management network node that sends a configuration packet from the central management net-

work node to the first node, wherein the configuration packet includes one or more synchronization configuration commands;

in the first node, logic that carries out the one or more synchronization configuration commands, and generates therefrom an updated synchronization status for the first node;

logic that sends a "Sync Link State Update" packet from the first node to the central management network node, wherein the "Sync Link State Update" packet includes the updated synchronization status for the first node; and

logic in the central management network node that sends a "Sync Link State Acknowledge" packet from the central management network node to the first node acknowledging receipt of the updated synchronization status for the first node in the "Sync Link State Update" packet.

23. An apparatus for updating a database in a first node in a synchronization network, the apparatus comprising:

logic in the first node that populates the database in the first node with synchronization information for the first node;

logic that sends a "Sync Hello" packet from the first node to a second node in the synchronization network, said "Sync Hello" packet including information about the synchronization status of the first node;

logic in the second node that updates a database in the second node with the information about the synchronization status of the first node and sends a first "Sync Database Description" packet from the second node to the first node in response to receipt of the "Sync Hello" packet, wherein the first "Sync Database Description" packet includes information about the synchronization status of the second node; and

in the first node, logic that updates the database with the information about the synchronization status of the second node.

24. The apparatus of claim 23, further comprising:

logic in the first node that acknowledges receipt of the first "Sync Database Description" packet in the first node by sending a second "Sync Database Description" packet from the first node to the second node.

* * * * *

EXHIBIT B

US006891656B2

(12) **United States Patent**
Hunt

(10) **Patent No.:** **US 6,891,656 B2**
(45) **Date of Patent:** **May 10, 2005**

(54) **PIXEL BASED GOBO RECORD CONTROL FORMAT**

(75) Inventor: **Mark Hunt**, Derby (GB)

(73) Assignee: **Production Resource Group, L.L.C.**, New Windsor, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **10/638,124**

(22) Filed: **Aug. 8, 2003**

(65) **Prior Publication Data**

US 2004/0061926 A1 Apr. 1, 2004

Related U.S. Application Data

(63) Continuation of application No. 10/271,521, filed on Oct. 15, 2002, which is a continuation of application No. 09/882,755, filed on Jun. 15, 2001, now Pat. No. 6,466,357, which is a continuation of application No. 09/500,393, filed on Feb. 8, 2000, now Pat. No. 6,256,136, which is a continuation of application No. 09/145,314, filed on Aug. 31, 1998, now Pat. No. 6,057,958.

(60) Provisional application No. 60/059,161, filed on Sep. 17, 1997, and provisional application No. 60/065,133, filed on Nov. 12, 1997.

(51) **Int. Cl.**[7] **G02B 26/00**; G06T 11/20; G06K 9/00; G06K 9/48; G03B 21/00

(52) **U.S. Cl.** **359/291**; 345/441; 345/419; 345/473; 345/604; 345/611; 345/619; 345/764; 382/181; 382/199; 382/192; 382/203; 382/154; 382/167; 353/122; 362/282

(58) **Field of Search** 359/291; 345/431, 345/611, 601, 604, 441, 473, 764, 419, 619, 589; 382/181, 192, 203, 154, 167; 353/15, 25, 122; 362/282, 284, 294

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,891,738 A	*	1/1990	Richardson et al.	362/282
5,113,332 A	*	5/1992	Richardson	362/282
5,282,121 A	*	1/1994	Bornhorst et al.	362/294
5,828,485 A		10/1998	Hewlett	359/291
5,852,443 A		12/1998	Kenworthy	345/431
6,256,136 B1		7/2001	Hunt	359/291
6,486,880 B2		11/2002	Clark-Schreyer et al.	345/473

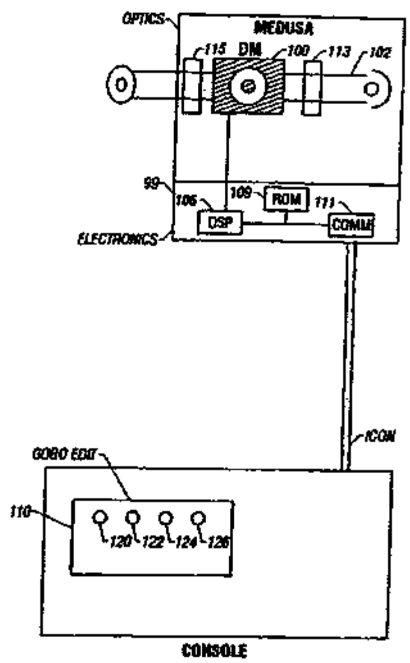
* cited by examiner

Primary Examiner—Loha Ben
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A special record format used for commanding light pattern shapes and addressable light pattern shape generator. The command format includes a first part which commands a specified gobo and second parts which command the characteristics of that gobo. The gobo is formed by making a default gobo based on the type and modifying that default gobo to fit the characteristics.

27 Claims, 6 Drawing Sheets



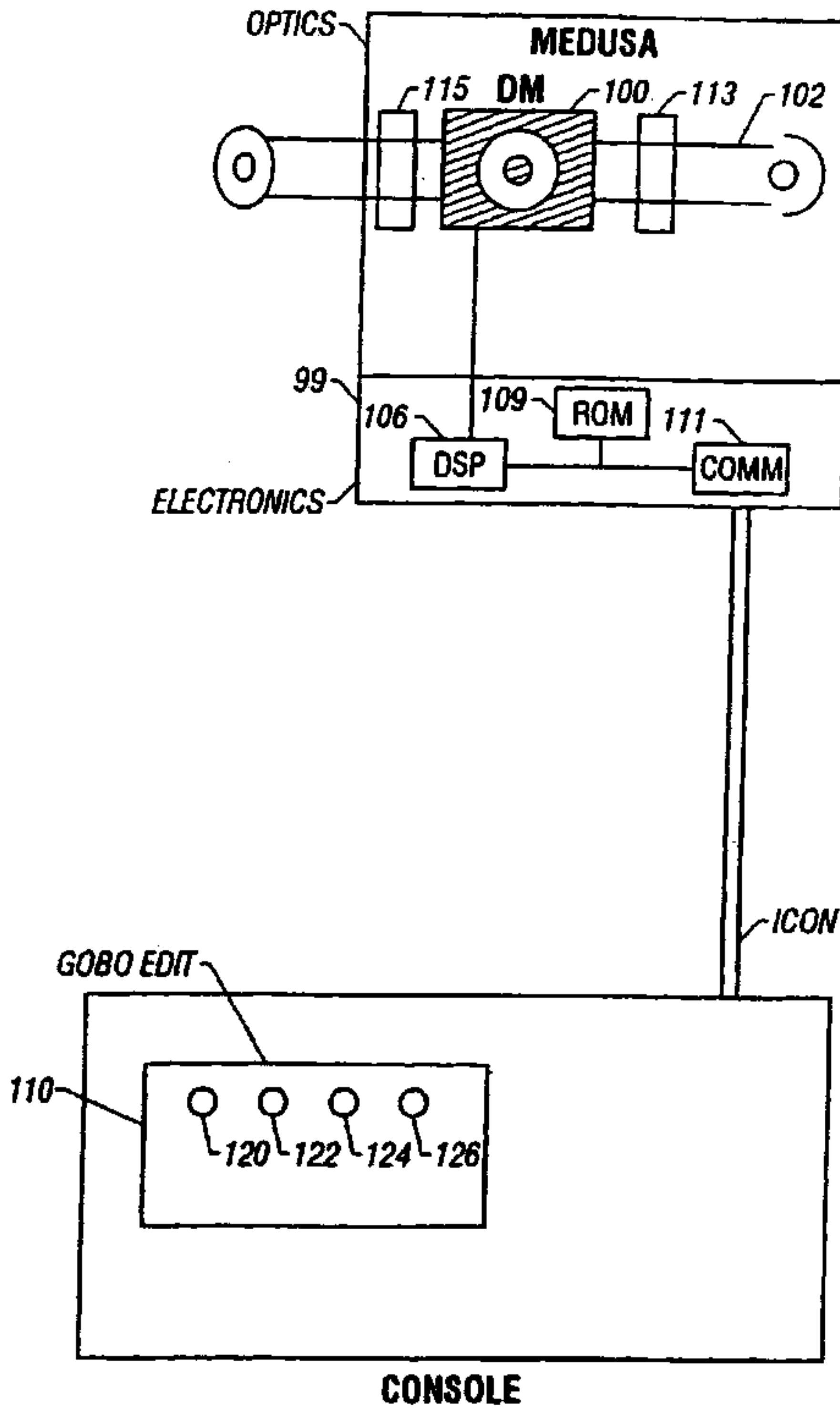

OPTICS
MEDUSA
115
DM
100
113
102
99
106
109
ROM
111
DSP
COMM
ELECTRONICS
ICON
GOBO EDIT
110
120
122
124
126
CONSOLE

FIG. 1

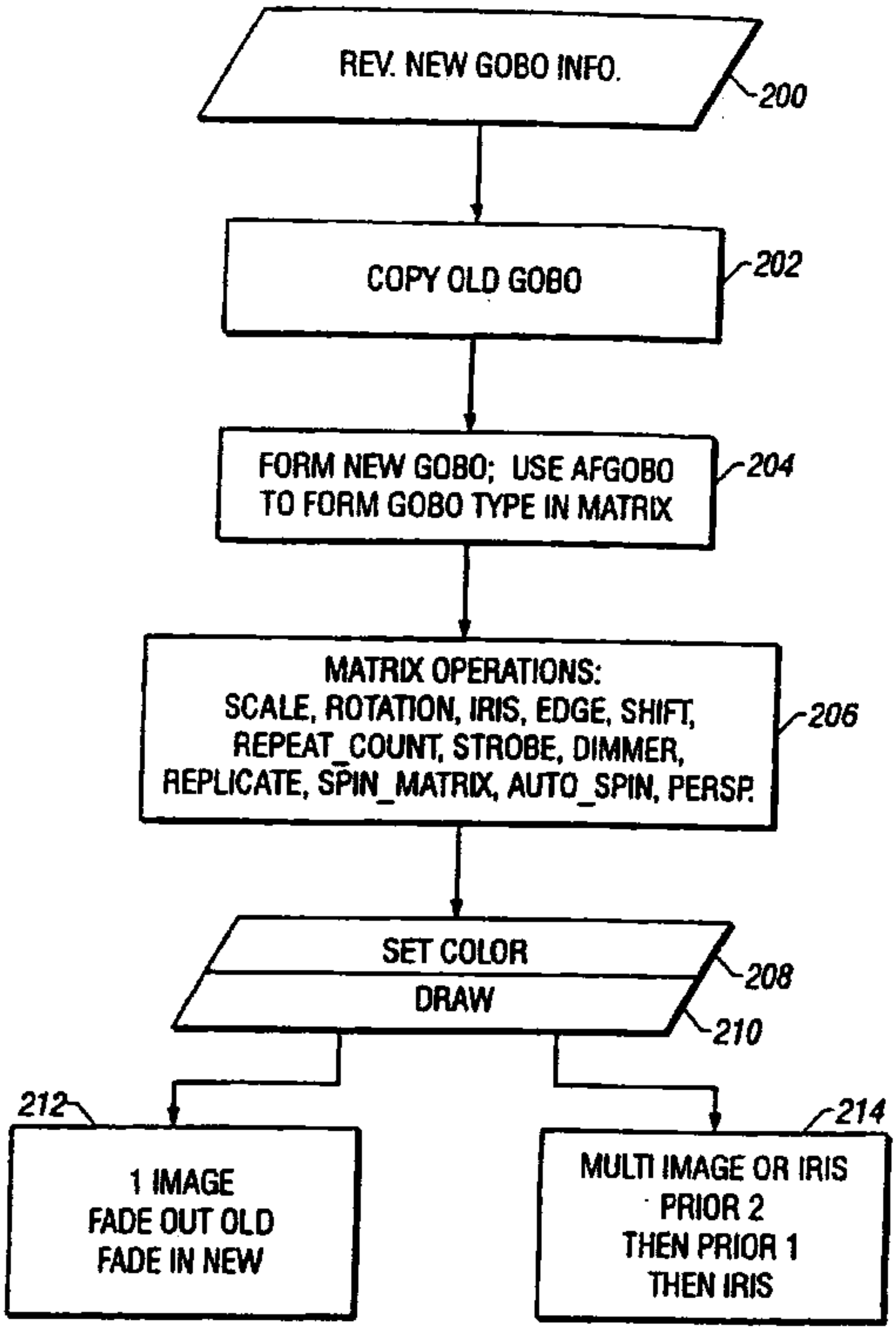

REV. NEW GOBO INFO.
200
COPY OLD GOBO
202
FORM NEW GOBO; USE AFGOBO TO FORM GOBO TYPE IN MATRIX
204
MATRIX OPERATIONS: SCALE, ROTATION, IRIS, EDGE, SHIFT, REPEAT_COUNT, STROBE, DIMMER, REPLICATE, SPIN_MATRIX, AUTO_SPIN, PERSP.
206
SET COLOR
208
DRAW
210
212
1 IMAGE FADE OUT OLD FADE IN NEW
214
MULTI IMAGE OR IRIS PRIOR 2 THEN PRIOR 1 THEN IRIS

FIG. 2

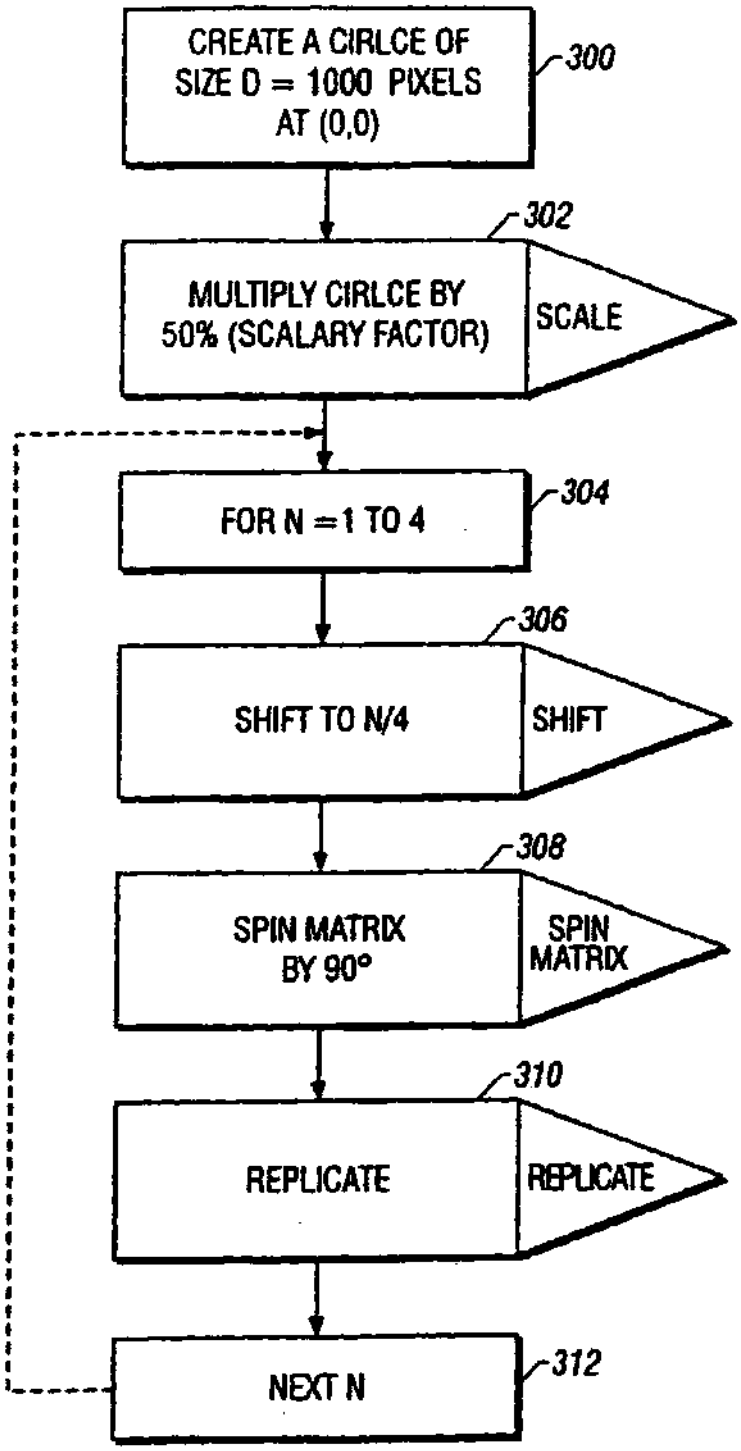

CREATE A CIRLCE OF SIZE D = 1000 PIXELS AT (0,0)
300
302
MULTIPLY CIRLCE BY 50% (SCALARY FACTOR)
SCALE
FOR N =1 TO 4
304
306
SHIFT TO N/4
SHIFT
308
SPIN MATRIX BY 90°
SPIN MATRIX
310
REPLICATE
REPLICATE
NEXT N
312

FIG. 3

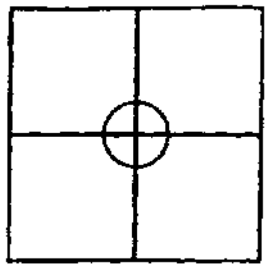


FIG. 4A

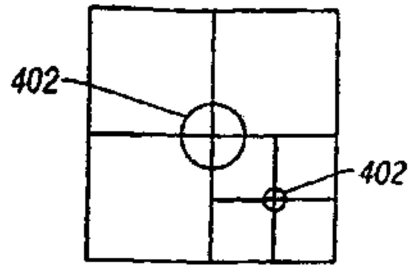


FIG. 4E

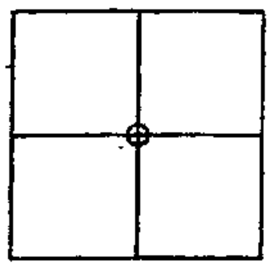


FIG. 4B

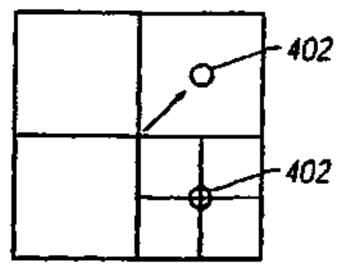


FIG. 4F

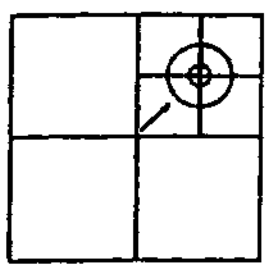


FIG. 4C

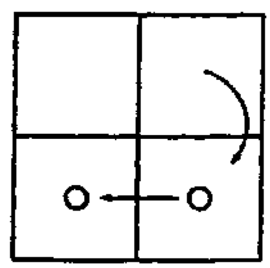


FIG. 4G

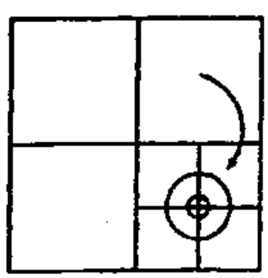


FIG. 4D

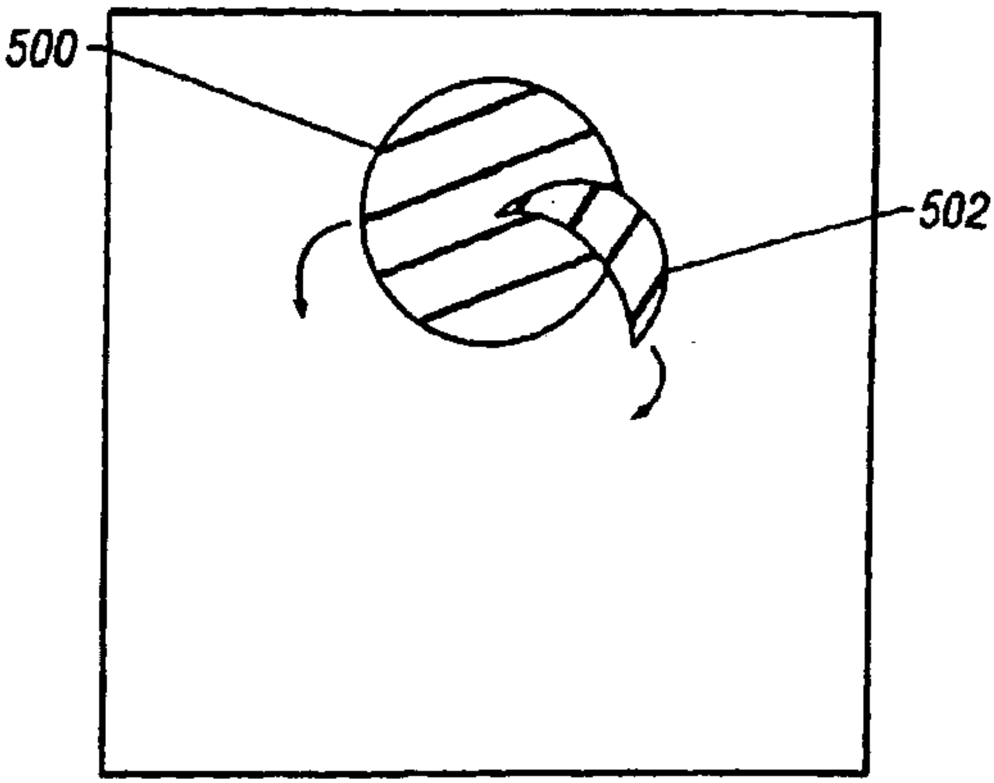


FIG. 5

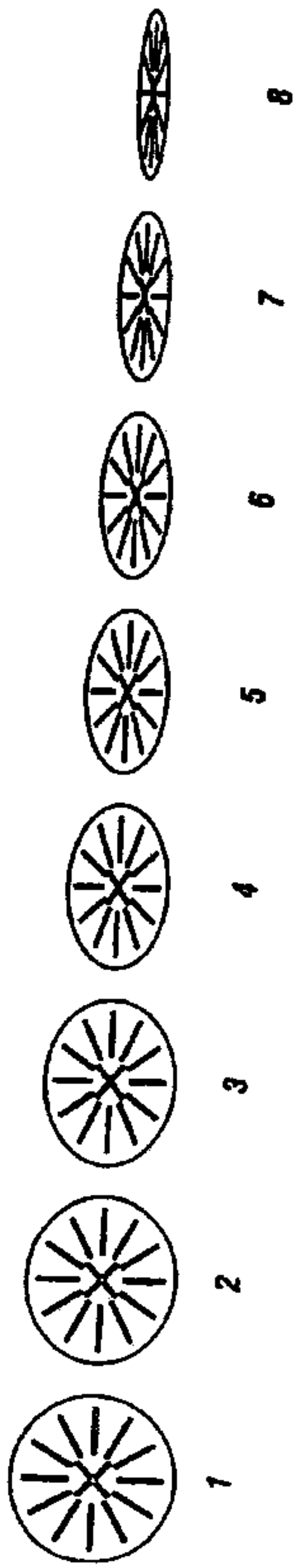


FIG. 6

PIXEL BASED GOBO RECORD CONTROL FORMAT

This application is a continuation of U.S. application Ser. No. 10/271,521, filed Oct. 15, 2002, which is a continuation of U.S. application Ser. No. 09/882,755, filed Jun. 15, 2001 now U.S. Pat. No. 6,466,357, which is a continuation of U.S. application Ser. No. 09/500,393, filed Feb. 8, 2000 now U.S. Pat. 6,256,136, which is a continuation of U.S. application Ser. No. 09/145,314, filed Aug. 31, 1998 now U.S. Pat. No. 6,057,958, which claims priority from U.S. Provisional application Nos. 60/059,161, filed Sep. 17, 1997, and 60/065,133, filed Nov. 12, 1997.

FIELD

The present invention relates to a system of controlling light beam pattern ("gobo") shape in a pixilated gobo control system.

BACKGROUND

Commonly assigned patent application Ser. No. 08/854,353, the disclosure of which is herewith incorporated by reference, describes a stage lighting system which operates based on computer-provided commands to form special effects. One of those effects is control of the shape of a light pattern that is transmitted by the device. This control is carried out on a pixel-by-pixel basis, hence referred to in this specification as pixilated. Control is also carried out using an x-y controllable device. The preferred embodiment describes using a digital mirror device, but other x-y controllable devices such as a grating light valve, are also contemplated.

The computer controlled system includes a digital signal processor **106** which is used to create an image command. That image command controls the pixels of the x-y controllable device to shape the light that it is output from the device.

The system described in the above-referenced application allows unparalleled flexibility in selection of gobo shapes and movement. This opens an entirely new science of controlling gobos. The present inventors found that, unexpectedly, even more flexibility is obtained by a special control language for controlling those movements.

SUMMARY

The present disclosure defines a way of communicating with an x-y controllable device to form special electronic light pattern shapes. More specifically, the present application describes using a control language to communicate with an electronic gobo in order to reposition part or all of the image that is shaping the light.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects of the invention will now be described with reference to the attached drawings, in which:

FIG. 1 shows a block diagram of the basic system operating the embodiment;

FIG. 2 shows a basic flowchart of operation;

FIG. 3 shows a flowchart of forming a replicating circles type gobo;

FIGS. 4A through 4G show respective interim results of carrying out the replicating circles operation;

FIG. 5 shows the result of two overlapping gobos rotating in opposite directions; and

FIGS. 6(1) through 6(8) show a z-axis flipping gobo.

DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 shows a block diagram of the hardware used according to the preferred embodiment. As described above, this system uses a digital mirror device **100**, which has also been called a digital mirror device ("DMD") and a digital light processor device ("DLP"). More generally, any system which allows controlling shape of light on a pixel basis, including a grating light valve, could be used as the light shaper. This light shaper forms the shape of light which is transmitted. FIG. 1 shows the light being transmitted as **102**, and shows the transmitted light. The information for the digital mirror **100** is calculated by a digital signal processor **106**. Information is calculated based on local information stored in the lamp, e.g., in ROM **109**, and also in information which is received from the console **104** over the communication link.

The operation is commanded according to a format.

The preferred data format provides 4 bytes for each of color and gobo control information.

The most significant byte of gobo control data, ("dfGobo") indicates the gobo type. Many different gobo types are possible. Once a type is defined, the gobo formed from that type is represented by a number. That type can be edited using a special gobo editor described herein. The gobo editor allows the information to be modified in new ways, and forms new kinds of images and effects.

The images which are used to form the gobos may have variable and/or moving parts. The operator can control certain aspects of these parts from the console via the gobo control information. The type of gobo controls the gobo editor to allow certain parameters to be edited.

The examples given below are only exemplary of the types of gobo shapes that can be controlled, and the controls that are possible when using those gobo shapes. Of course, other controls of other shapes are possible and predictable based on this disclosure.

A first embodiment is the control of an annulus, or "ring" gobo. The DMD **100** in FIG. 1 is shown with the ring gobo being formed on the DMD. The ring gobo is type 000A. When the gobo type 0A is enabled, the gobo editor **110** on the console **104** is enabled and the existing gobo encoders **120**, **122**, **124**, and **126** are used. The gobo editor **110** provides the operator with specialized control over the internal and the external diameters of the annulus, using separate controls in the gobo editor.

The gobo editor and control system also provides other capabilities, including the capability of timed moves between different edited parameters. For example, the ring forming the gobo could be controlled to be thicker. The operation could then effect a timed move between these "preset" ring thicknesses. Control like this cannot even be attempted with conventional fixtures.

Another embodiment is a composite gobo with moving parts. These parts can move though any path that are programmed in the gobo data itself. This is done in response to the variant fields in the gobo control record, again with timing. Multiple parts can be linked to a single control allowing almost unlimited effects.

Another embodiment of this system adapts the effect for an "eye" gobo, where the pupil of the eye changes its position (look left, look right) in response to the control.

Yet another example is a Polygon record which can be used for forming a triangle or some other polygonal shape.

The control can be likened to the slider control under a QuickTime movie window, which allows you to manually move to any point in the movie. However, our controls need not be restricted to timelines.

Even though such moving parts are used, scaling and rotation on the gobo is also possible.

The following type assignments are contemplated:

00–0F=FixedGobo (with no "moving parts")
10–1F=SingleCntrl (with 1 "moving part")
20–2F=DoubleCntrl (with 2 "moving parts")
30–FF=undefined, reserved.

The remaining control record bytes for each type are defined as follows:

Byte: memory	dfGobo2	dfGobo3	dfGobo4	#gobos/ type.	total
FixedGobo:	ID[23:16]	ID[15:8]	ID[7:0]	16 M/type	256 M
SingleCntrl:	ID[15:8]	ID[7:0]	control#1	64 k/type	1 M
DoubleCntrl:	ID[7:0]	control#2	control#1	256/type	4 k

As can be seen from this example, this use of the control record to carry control values does restrict the number of gobos which can be defined of that type, especially for the 2-control type.

Console Support:

The use of variant part gobos requires no modifications to existing console software for the ICON (7M) console. The Gobo editor in current ICON software already provides 4 separate encoders for each gobo. These translate directly to the values of the 4 bytes sent in the communications data packet as follows:

Byte: dfGobo Enc: TopRight	dfGobo2 MidRight	dfGobo3 BotRight	dfGobo4 BotLeft
FixedGobo:	ID[23:16]	ID[15:8]	ID[7:0]
SingleCntrl:	ID[15:8]	ID[7 0]	control#1
DoubleCntrl:	ID[7:0]	control#2	control#1

These values would be part of a preset gobo, which could be copied as the starting point.

Once these values are set, the third and fourth channels automatically become the inner/outer radius controls. Using two radii allows the annulus to be turned "inside out".

Each control channel's data always has the same meaning within the console. The console treats these values as simply numbers that are passed on. The meanings of those numbers, as interpreted by the lamps change according to the value in dfGobo.

The lamp will always receives all 4 bytes of the gobo data in the same packet. Therefore, a "DoubleCntrl" gobo will always have the correct control values packed along with it.

Hence, the console needs no real modification. If a "soft" console is used, then name reassignments and/or key reassignments may be desirable.

Timing:

For each data packet, there is an associated "Time" for gobo response. This is conventionally taken as the time allotted to place the new gobo in the light gate. This delay has been caused by motor timing. In this system, variant gobo, the control is more dynamically used. If the non-variant parts of the gobo remain the same, then it is still the same gobo, only with control changes. Then, the time value is interpreted as the time allowed for the control change.

Since different gobo presets (in the console) can reference the same gobo, but with different control settings, this allows easily programmed timed moves between different annuli, etc.

Internal Workings:

When the gobo command data is extracted from the packet at the lamp, the dfGobo byte is inspected first, to see if either dfGobo3 or dfGobo4 are significant in selecting the image. In the case of the "Cntrl" variants, one or both of these bytes is masked out, and the resulting 32-bit number is used to search for a matching gobo image (by Gobo_1D) in the library stored in the lamp's ROM **109**.

If a matching image is found, and the image is not already in use, then the following steps are taken:

1) The image data is copied into RAM, so that its fields may be modified by the control values. This step will be skipped if the image is currently active.

2) The initial control values are then recovered from the data packet, and used to modify certain fields of the image data, according to the control records.

3) The image is drawn on the display device, using the newly-modified fields in the image data.

If the image is already in use, then the RAM copy is not altered. Instead, a time-sliced task is set up to slew from the existing control values to those in the new data packet, in a time determined by the new data packet.

At each vertical retrace of the display, new control values are computed, and steps 2 (using the new control values) and 3 above are repeated, so that the image appears modified with time.

The Image Data Records:

All images stored in the lamp are in a variant record format:

Header:

Length 32 bits, offset to next gobo in list.
Gobo _1D 32 bits, serial number of gobo.
Gobo records:

Length	32 bits, offset to next record.
Opcode	16 bits, type of object to be drawn.
Data	Variant part - data describing object.
_Length	32 bits, offset to next record.
Opcode	16 bits, type of object to be drawn.
Data	Variant part - data describing object.

_ EndMarker 64 bits, all zeroes - indicates end of gobo data.
+ Next gobo, or End Marker, indicating end of gobo list.

Gobos with controls are exactly the same, except that they contain control records, which describe how the control values are to affect the gobo data. Each control record contains the usual length and Opcode fields, and a field containing the control number (1 or 2).

These are followed by a list of "field modification" records. Each record contains information about the offset (from the start of the gobo data) of the field, the size (8, 16 or 32 bits) of the field, and how its value depends on the control value.

Length	32 bits, offset to next record
Opcode	16 bits = control_record (constant)

-continued

CntrlNum	16 bits = 1 or 2 (control number)
	/* field modification record #1 */
Address	16 bits, offset from start of gobo to affected field.
Flags	16 bits, information about field (size, signed, etc)
Scale	16 bits, scale factor applied to control before use
ZPoint	16 bits, added to control value after scaling.
	/* field modification record #2 */
Address	16 bits, offset from start of gobo to affected field.
Flags	16 bits, information about field (size, signed, etc)
Scale	16 bits, scale factor applied to control before use
ZPoint	16 bits, added to control value after scaling.

As can be seen, a single control can have almost unlimited effects on the gobo, since ANY values in the data can be modified in any way, and the number of field modification records is almost unlimited.

Note that since the control records are part of the gobo data itself, they can have intimate knowledge of the gobo structure. This makes the hard-coding of field offsets acceptable.

In cases where the power offered by this simple structure is not sufficient, a control record could be defined which contains code to be executed by the processor. This code would be passed parameters, such as the address of the gobo data, and the value of the control being adjusted.

EXAMPLE RECORDS

The Annulus record has the following format:

Length	32 bits
Opcode	16 bits, = type_annulus
Pad	16 bits, unused
Centre_x	16 bits, x coordinate of centre
Centre_y	16 bits, y coordinate of centre
OuterRad	16 bits, outside radius (the radii get swapped when drawn if their values are in the wrong order)
InnerRad	16 bits, inside radius

It can be seen from this that it is easy to "target" one of the radius parameters from a control record. Use of two control records, each with one of the radii as a target, would provide full control over the annulus shape.

Note that if the centre point coordinates are modified, the annulus will move around the display area, independent of any other drawing elements in the same gobo's data.

The Polygon record for a triangle has this format:

Length	32 bits
Opcode	16 bits, = type_polygon
Pad	16 bits, vertex count = 3
Centre_x	16 bits, x coordinate of vertex
Centre_y	16 bits, y coordinate of vertex
Centre_x	16 bits, x coordinate of vertex
Centre_y	16 bits, y coordinate of vertex
Centre_x	16 bits, x coordinate of vertex
Centre_y	16 bits, y coordinate of vertex

It is easy to modify any of the vertex coordinates, producing distortion of the triangle.

The gobo data can contain commands to modify the drawing environment, by rotation, scaling, offset, and color control, the power of the control records is limitless.

Second Embodiment

This second embodiment provides further detail about implementation once the gobo information is received.

Gobo information is, at times, being continuously calculated by DSP **106**. The flowchart of FIG. **2** shows the handling operation that is carried out when new gobo information is received.

At step **200**, the system receives new gobo information. In the preferred embodiment, this is done by using a communications device ill in the lamp **99**. The communications device is a mailbox which indicates when new mail is received. Hence, the new gobo information is received at step **200** by determining that new mail has been received.

At step **202**, the system copies the old gobo and switches pointers. The operation continues using the old gobo until the draw routine is called later on.

At step **204**, the new information is used to form a new gobo. The system uses a defined gobo ("dfGobo") as discussed previously which has a defined matrix. The type dfGobo is used to read the contents from the memory **109** and thereby form a default image. That default image is formed in a matrix. For example, in the case of an annulus; a default size annulus can be formed at position **0,0** in the matrix. An example of forming filled balls is provided herein.

Step **206** represents calls to subroutines. The default gobo is in the matrix, but the power of this system is its ability to very easily change the characteristics of that default gobo. In this embodiment, the characteristics are changed by changing the characteristics of the matrix and hence, shifting that default gobo in different ways. The matrix operations, which are described in further detail herein, include scaling the gobo, rotation, iris, edge, strobe, and dimmer. Other matrix operations are possible. Each of these matrix operations takes the default gobo, and does something to it.

For example, scale changes the size of the default gobo rotation rotates the default gobo by a certain amount.

Iris simulates an iris operation by choosing an area of interest, typically circular, and erasing everything outside that area of interest. This is very easily done in the matrix, since it simply defines a portion in the matrix where all black is written.

Edge effects carry out certain effects on the edge is such as softening the edge. This determines a predetermined thickness, which is translated to a predetermined number of pixels, and carries out a predetermined operation on the number of pixels. For example, for a 50% edge softening, every other pixel can be turned off. The strobe is in effect that allows all pixels to be turned on and off at a predetermined frequency, i.e., 3 to 10 times a second. The dimmer allows the image to be made dimmer by turning off some of the pixels at predetermined times.

The replicate command forms another default gobo, to allow two different gobos to be handled by the same record. This will be shown with reference to the exemplary third embodiment showing balls. Each of those gobos is then handled as the same unit and the entirety of the gobos can be, for example, rotated. The result of step **206** and all of these subroutines that are called is that the matrix includes information about the bits to be mapped to the digital mirror **100**.

At step **208**, the system then obtains the color of the gobos from the control record discussed previously. This gobo color is used to set the appropriate color changing circuitry

113 and 115 in the lamp 99. Note that the color changing circuitry is shown both before and after the digital mirror 100. It should be understood that either of those color changing circuits could be used by itself.

At step 210, the system calls the draw routine in which the matrix is mapped to the digital mirror. This is done in different ways depending on the number of images being used. Step 212 shows the draw routine for a single image being used as the gobo. In that case, the old gobo, now copied as shown in step 202, is faded out while the new gobo newly calculated is faded in. Pointers are again changed so that the system points to the new gobo. Hence, this has the effect of automatically fading out the old gobo and fading in the new gobo.

Step 214 schematically shows the draw routine for a system with multiple images for an iris. In that system, one of the gobos is given priority over the other. If one is brighter than the other, then that one is automatically given priority. The one with priority 2, the lower priority one, is written first. Then the higher priority gobo is written. Finally, the iris is written which is essentially drawing black around the edges of the screen defined by the iris. Note that unlike a conventional iris, this iris can take on many different shapes. The iris can take on not just a circular shape, but also an elliptical shape, a rectangular shape, or a polygonal shape. In addition, the iris can rotate when it is non-circular so that for the example of a square iris, the edges of the square can actually rotate.

Returning to step 206, in the case of a replicate, there are multiple gobos in the matrix. This allows the option of spinning the entire matrix, shown as spin matrix.

An example will now be described with reference to the case of repeating circles. At step 200, the new gobo information is received indicating a circle. This is followed by the other steps of 202 where the old gobo is copied, and 204 where the new gobo is formed. The specific operation forms a new gobo at step 300 by creating a circle of size diameter equals 1000 pixels at origin 00. This default circle is automatically created. FIG. 4A shows the default gobo which is created, a default size circle at 00. It is assumed for purposes of this operation that all of the circles will be the same size.

At step 302, the circle is scaled by multiplying the entire circle by an appropriate scaling factor. Here, for simplicity, we are assuming a scaling factor of 50% to create a smaller circle. The result is shown in FIG. 4B. A gobo half the size of the gobo of FIG. 4A is still at the origin. This is actually the scale of the subroutine as shown in the right portion of step 302. Next, since there will be four repeated gobos in this example, a four-loop is formed to form each of the gobos at step 304. Each of the gobos is shifted in position by calling the matrix operator shift. In this example, the gobo is shifted to a quadrant to the upper right of the origin. This position is referred to as π over 4 in the FIG. 3 flowchart and results in the gobo being shifted to the center portion of the top right quadrant as shown in FIG. 4C. This is again easily accomplished within the matrix by moving the appropriate values. At step 308, the matrix is spun by 90 degrees in order to put the gobo in the next quadrant as shown in FIG. 4D in preparation for the new gobo being formed into the same quadrant. Now the system is ready for the next gobo, thereby calling the replicate command which quite easily creates another default gobo circle and scales it. The four-loop is then continued at step 312.

The replicate process is shown in FIG. 4E where a new gobo 402 is formed in addition to the existing gobo 400. The system then passes again through the four-loop, with the results being shown in the following figures. In FIG. 4F, the new gobo 402 is again moved to the upper right quadrant (step 306). In FIG. 4G, the matrix is again rotated to leave room for a new gobo in the upper right quadrant. This continues until the end of the four-loop. Hence, this allows each of the gobos to be formed.

Since all of this is done in matrix operation, it is easily programmable into the digital signal processor. While the above has given the example of a circle, it should be understood that this scaling and moving operation can be carried out for anything. The polygons, circles, annulus, and everything else is easily scaled.

The same operation can be carried out with the multiple parameter gobos. For example, for the case of a ring, the variable takes the form annulus (inner R, outer R, x and y). This defines the annulus and turns of the inner radius, the outer radius, and x and y offsets from the origin. Again, as shown in step 3, the annulus is first written into the matrix as a default size, and then appropriately scaled and shifted. In terms of the previously described control, the ring gobo has two controls: control 1 and control 2 defined the inner and outer radius.

Each of these operations is also automatically carried out by the command repeat count which allows easily forming the multiple position gobo of FIGS. 4A–4G. The variable auto spin defines a continuous spin operation. The spin operation commands the digital signal processor to continuously spin the entire matrix by a certain amount each time.

One particularly interesting feature available from the digital mirror device is the ability to use multiple gobos which can operate totally separately from one another raises the ability to have different gobos spinning in different directions. When the gobos overlap, the processor can also calculate relative brightness of the two gobos. In addition, one gobo can be brighter than the other. This raises the possibility of a system such as shown in FIG. 5. Two gobos are shown spinning in opposite directions: the circle gobo 500 is spinning the counterclockwise direction, while the half moon gobo 502 is spinning in the clockwise direction. At the overlap, the half moon gobo which is brighter than the circle gobo, is visible over the circle gobo. Such effects were simply not possible with previous systems. Any matrix operation is possible, and only a few of those matrix operations have been described herein.

A final matrix operation to be described is the perspective transformation. This defines rotation of the gobo in the Z axis and hence allows adding depth and perspective to the gobo. For each gobo for which rotation is desired, a calculation is preferably made in advance as to what the gobo will look like during the Z axis transformation. For example, when the gobo is flipping in the Z axis, the top goes back and looks smaller while the front comes forward and looks larger. FIGS. 6(1)–6(8) show the varying stages of the gobo flipping.

In FIG. 6(8), the gobo has its edge toward the user. This is shown in FIG. 6(8) as a very thin line, e.g., three pixels wide, although the gobo could be zero thickness at this point.

Automatic algorithms are available for such Z axis transformation, or alternatively a specific Z axis transformation can be drawn and digitized automatically to enable a custom look.

Although only a few embodiments have been described in detail above, other embodiments are contemplated by the inventor and are intended to be encompassed within the following claims. In addition, other modifications are contemplated and are also intended to be covered.

What is claimed is:

1. A system, comprising:

a console, having at least one file therein which is indicative of an image, at least one control therein which controls some aspect of a light pattern shape based on the image, and a processing element that produces an output signal defining a record that includes information about said light pattern shape and at least one parameter representing said light pattern shape.

2. A system as in claim 1, wherein said console includes a digital signal processor which computes information related to said output signal.

3. A system as in claim 1, wherein said control allows controlling a size of the light pattern shape.

4. A system as in claim 1, wherein said control allows controlling e basic shape and editing the shape.

5. A system as in claim 1, wherein said control allows controlling a color of the shaped light.

6. A system as in claim 1, wherein said record defines an output signal that controls a light.

7. A system as in claim 1, wherein said record includes a first portion defining a gobo type, and a second portion defining characteristics of said gobo type.

8. A system as in claim 7, wherein said characteristics include size of the gobo as defined by the first portion.

9. A system, comprising:

a console, having at least one control thereon, which stores images, allows selection of one of said images, to define a gobo, and includes a processing element which produces an output signal indicative of the gobo defined by said one of said images.

10. A system as in claim 9, further comprising a control on said console which allows controlling variation of the gobo, and which changes the output signal indicative of said control.

11. A system as in claim 10, wherein the output signal includes a first value indicative of the gobo type, and a second value indicative of said variation.

12. A system as in claim 9, wherein said output signal includes a first value indicative of the gobo type and a second value indicative of additional aspects of the gobo.

13. A system as in claim 12, wherein said first value is a number indicative of a gobo shape.

14. A system as in claim 13, wherein said second value is information indicative of gobo sides.

15. A system as in claim 13, wherein said second value is information indicative of gobo time.

16. A format, comprising, a first set of bits defining the gobo control data which represents a form of a digital gobo and a second set of bits defining color data which represents a color of the digital gobo.

17. A format as in claim 16, wherein said format includes another set of bits indicative of gobo size.

18. A format as in claim 16, wherein said form of the digital gobo includes information indicative of the shape.

19. A format as in claim 16, wherein said first set of bits includes a type assignment which indicates a number of moving parts.

20. A format comprising a first set of bits defining a gobo to be projected, and a second set of bits defining the timing of moving parts of the gobo.

21. A format as in claim 20, wherein said first set of bits defines at least a shape of the gobo.

22. A format as in claim 20, wherein said second set of bits defines timing of a movement of the gobo.

23. A format as in claim 20, wherein said second set of bits defines timing of a change in size of said gobo.

24. A format comprising a first set of bite defining a gobo to be projected, said first set of bits defining a format of a notification of said gobo, and at least one additional set of bits, whose significance is determined by the first set of bits.

25. A format as in claim 24, wherein the first set of bits defines a gobo type and includes a number of parameters of the gobo, and said at least one additional set of bits defines said parameters.

26. A method, comprising:

receiving, in a light projecting element, a signal representing a digital gobo, and

using said signal representing the digital gobo to alter the shape of projected light on a pixel by pixel basis.

27. A method as in claim 26, wherein said using comprises carrying out a matrix arithmetic operation using said signal.

* * * * *

EXHIBIT C

US006919916B2

(12) United States Patent
Perry

(10) Patent No.: US 6,919,916 B2
(45) Date of Patent: Jul. 19, 2005

(54) **METHOD AND DEVICE FOR CREATING A FACSIMILE OF AN IMAGE**

(75) Inventor: **Scott M Perry**, Altadena, CA (US)

(73) Assignee: **Production Resource Group Inc.**, New Windsor, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 138 days.

(21) Appl. No.: **09/837,876**

(22) Filed: **Apr. 17, 2001**

(65) **Prior Publication Data**

US 2002/0080411 A1 Jun. 27, 2002

Related U.S. Application Data

(63) Continuation of application No. 07/461,344, filed on Jan. 5, 1990, now Pat. No. 6,219,093.

(51) **Int. Cl.**[7] **H04N 7/18**

(52) **U.S. Cl.** **348/135**; 348/161; 348/169

(58) **Field of Search** 348/135, 143, 348/161, 169; 358/400, 496, 406, 488; H04N 7/18, 5/225

(56) **References Cited**

U.S. PATENT DOCUMENTS

2,777,011 A	*	1/1957	Marks	348/42
3,401,262 A	*	9/1968	Fergason	250/331
3,986,022 A	*	10/1976	Hyatt	250/205

FOREIGN PATENT DOCUMENTS

JP	5277699	*	6/1977

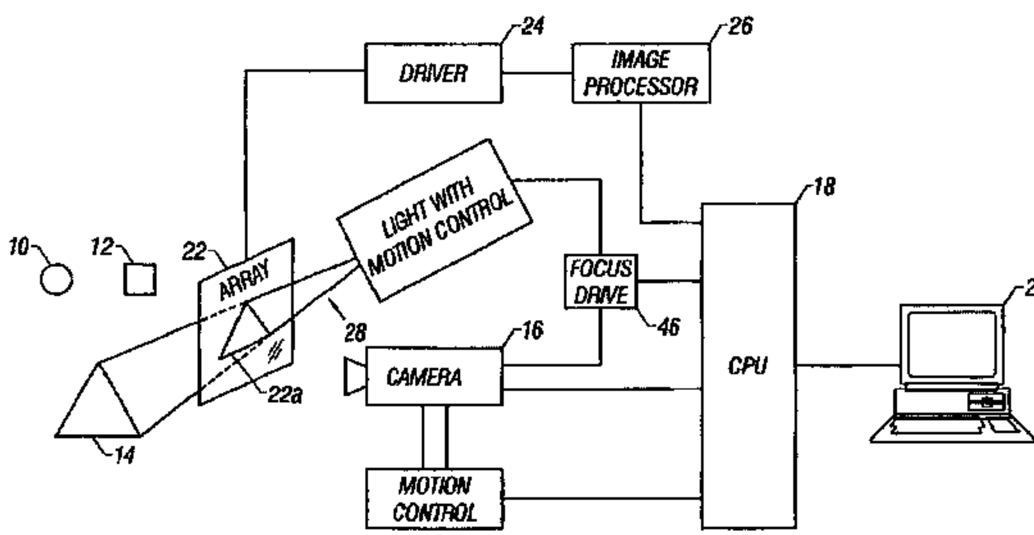
* cited by examiner

Primary Examiner—Sherrie Hsia
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A technique of using an electronic device to change light into the facsimile of an image. Signals are generated, indicative of the image shape. An array of electro optical devices are transformed based on the image signals, and those devices are used to transform the light characteristics.

1 Claim, 2 Drawing Sheets



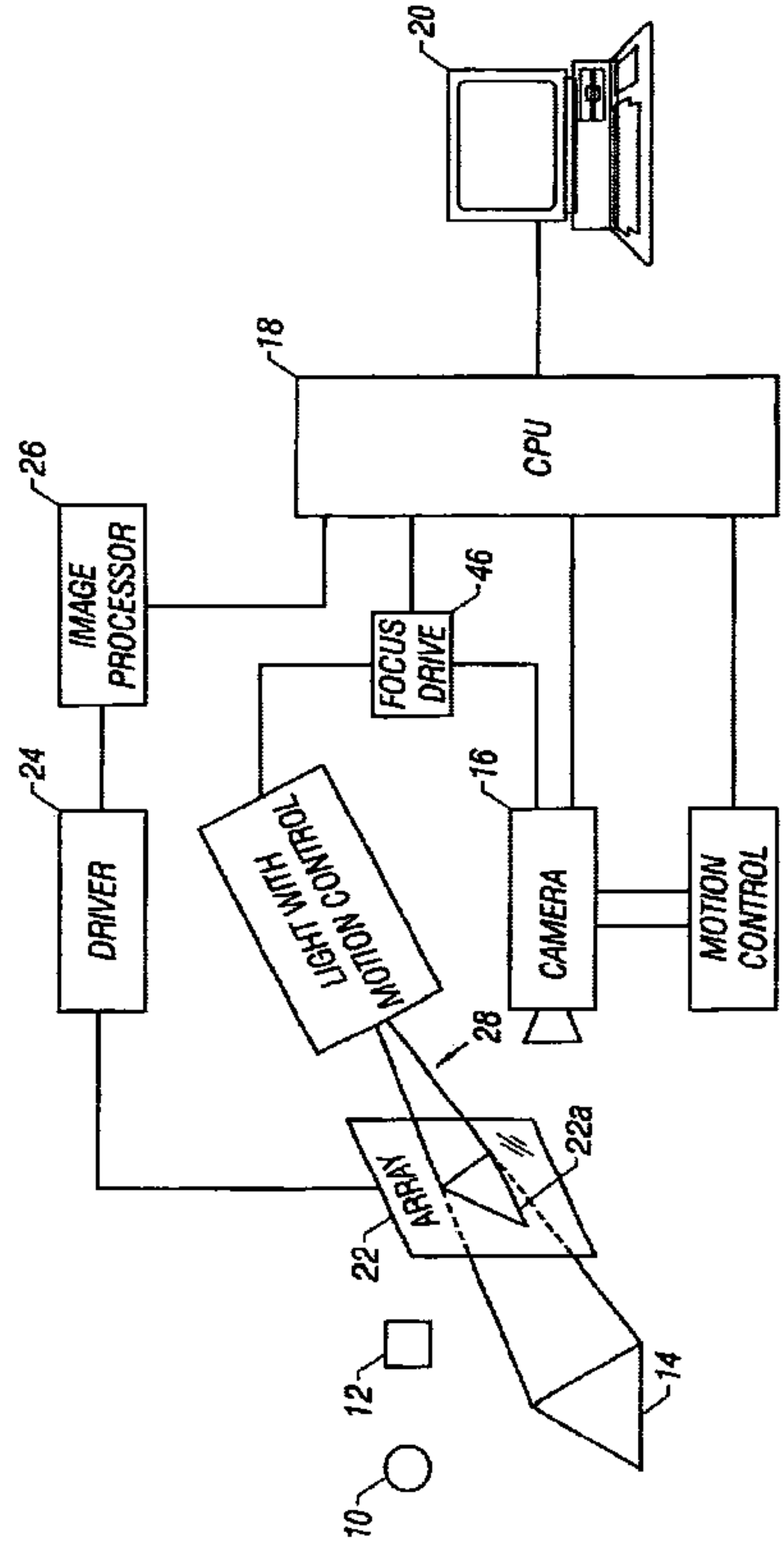

20
18
CPU
26
IMAGE PROCESSOR
46
FOCUS DRIVE
24
DRIVER
LIGHT WITH MOTION CONTROL
16
CAMERA
MOTION CONTROL
28
22a
ARRAY
22
12
14
10

FIG. 1

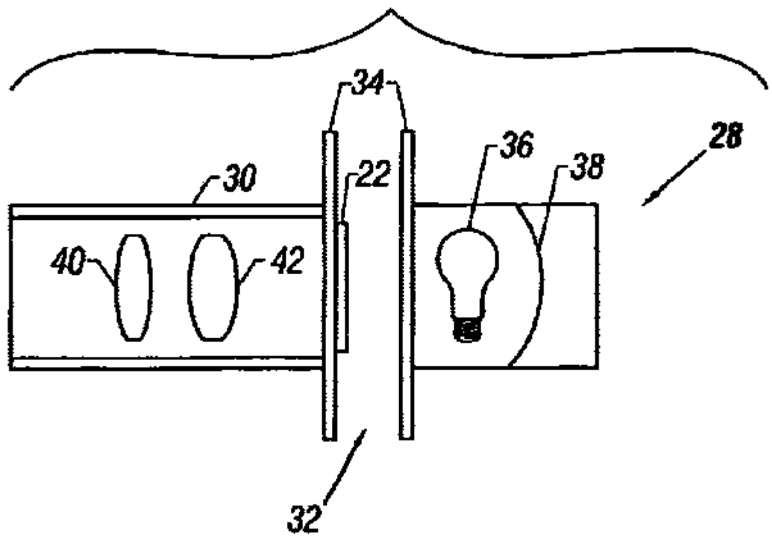

34
36
38
28
30
22
40
42
32

FIG. 2

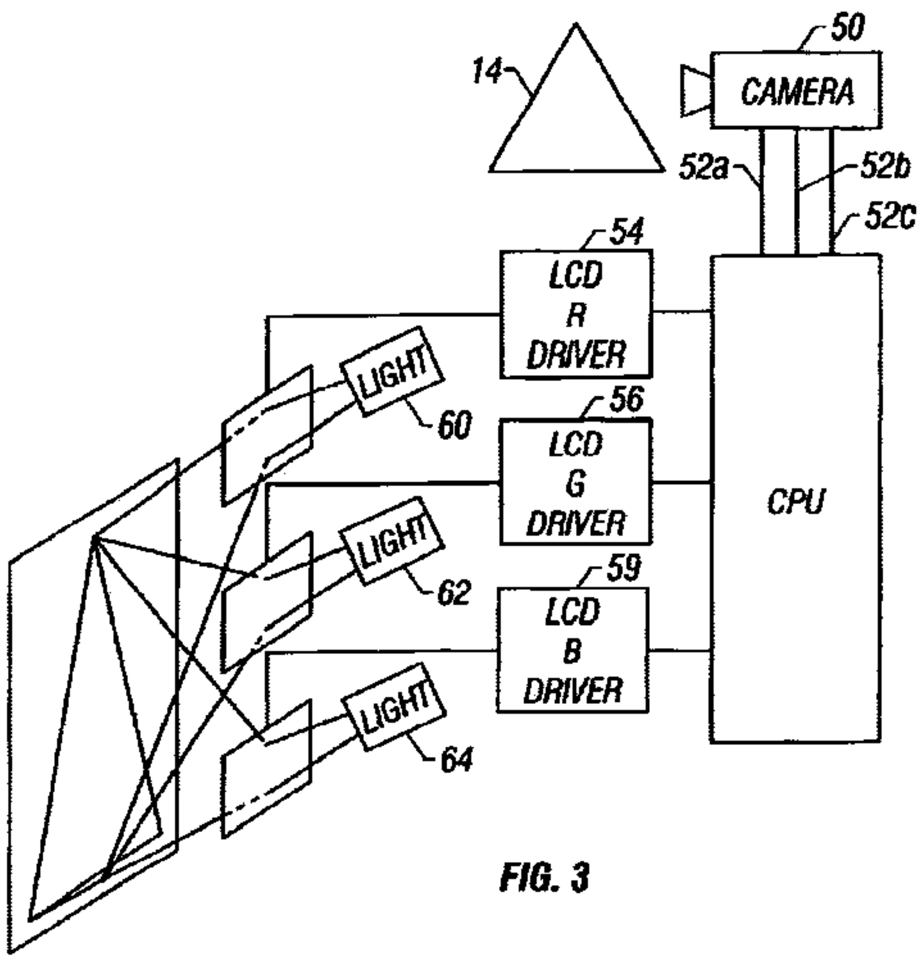

50
CAMERA
14
52a
52b
52c
54
LCD
R
DRIVER
LIGHT
60
56
LCD
G
DRIVER
CPU
LIGHT
62
59
LCD
B
DRIVER
LIGHT
64

FIG. 3

METHOD AND DEVICE FOR CREATING A FACSIMILE OF AN IMAGE

This is a continuation of U.S. application Ser. No. 07/461,344, filed Jan. 5, 1990 now U.S. Pat. No. 6,219,093 (allowed).

FIELD OF THE INVENTION

This invention relates to a device and method for creation of a facsimile of one or more selected images, which may or may not be moveable within a field of images. More particularly this invention relates to a method and device for selecting one or more images from a field, tracking the image, creating a facsimile of the image and continuously updating the facsimile as the image changes. Still more particularly and with reference to but one embodiment of the present invention a device and method are set forth for selecting an image, creating a real-time facsimile of the selected image at an opaque electro-optical device such as an LCD and projecting electromagnetic radiation including coherent or incoherent visible light through the device to illuminate the selected image and minimize the shadow created by the illumination.

BACKGROUND OF THE INVENTION

With reference to but one specific embodiment of the present invention, it has been long known to illuminate an entertainer on stage using one or more spotlights. As the entertainer moves and traverses the stage, the spotlights are manipulated to follow him or her. This illumination of the entertainer, while attenuated by the use of several spotlights, results in overspotting which creates shadows cast on the stage or backdrop. While the beams from the spotlights may be focused, illumination of the entertainer and his or her limbs creates the aforesaid shadows and spotting. These shadows can detract from the entertainer and may cause reflections or other problems during filming or videotaping. Heretofore there has been no satisfactory means to illuminate the movable entertainer so as to eliminate shadows and spotting. Further it has been necessary to provide lighting personnel at each spotlight to track the entertainer and focus the beam. This increases the labor costs for lighting and also restricts the placement of the spotlights in that sufficient space must be provided to accommodate the operator.

SUMMARY OF THE INVENTION

There is therefore provided according to but one aspect of the present invention a device and method to select one or several images (eg. an entertainer) from a field of images (eg. the stage and props) and to create a transparent facsimile or silhouette which changes in accordance to the movement of the entertainer. Illuminating light or other forms of radiation is projected and focused through the facsimile to illuminate only the form of the entertainer thereby eliminating or minimizing the casting of shadows and spotting. In a further aspect of the present invention, if the selected image is movable within the field, tracking is provided through image recognition and comparison or by an interactive locator or by pre-programming the movements of the image to correspond to predetermined movements and changes in form of the image. In a more broad aspect of the present invention, a device and method are set forth for creating a facsimile or silhouette of the image. For example in animation, an animal may be the selected image. Following the movements of the animal a facsimile or silhouette of the movements and form of the animal is created. By photographic or tracing or other techniques the facsimile may then be used to create an animated animal whose movements closely mimic those of the selected animal. In still a further aspect of the present invention, transparent facsimiles are created each for the colors red, green and blue of a color scene. By projecting red, green and blue light through their respective transparencies a full color image is thereby created upon a screen.

In still another aspect of the present invention x-ray or other forms of radiation may be directed through a suitable facsimile to thereby cast radiation from the source in a pattern dictated by the facsimile.

These and other feathers and advantages will become better appreciated as the same becomes better understood with reference to the drawings, specification and claims.

BRIEF DESCRIPTION OF THE DRAWING

With reference to these feathers and advantages the present invention will now be described with reference to the drawings wherein:

FIG. 1 depicts one embodiment of a device and method for creating a facsimile of an image within a field of images;

FIG. 2 is a side section view of a device for projecting an image of the aforesaid facsimile; and

FIG. 3 illustrates another embodiment of their present invention for recreating a color image.

DESCRIPTION

With reference to FIG. 1, one embodiment of the present invention will now be described. It is to be understood that the invention as shown and described is simplified to aid in the understanding thereof.

Accordingly a field of objects is provided which includes a circle **10**, square **12** and triangle **14** are perceived by camera **16**. It is to be understood that the field could be a stage including an entertainer, cast and props. Further it is to be understood that while the description hereinafter set forth relates primarily to images that are visually perceptible, that images may be perceptible at other electromagnetic ranges such as infrared or radar. In those instances, camera would be an infrared camera or a radar receiver. Further the camera as described according to the present invention may be a sonic receiver such as a sonar receiver. Suffice it to say, camera as used according to the present invention should be understood to be any type of device capable of receiving indicia of a field no matter by what medium the field is perceived.

Camera **16** may be a digital camera or a line scanning camera which typically scans 480 to 700 horizontal scan lines per frame or a charged coupled device.

Signals from camera **16** are arranged into a matrix of signals such as by the technique disclosed in U.S. Pat. No. 4,760,607 issued Jul. 26, 1988 to Sternberg et al. which is hereby incorporated by reference. Accordingly signals from camera **16** are processed by central processing unit or CPU **18** into a suitable matrix of pixel signal words, each representative of at least the intensity of a segment or point of the image field as perceived by camera **16**. For example each word may be the intensity of the point from a range of one to eight with eight being a maximum intensity and one being a minimum intensity as is described in Sternberg et al. referenced above.

Means are provided to, based in part upon the matrix of pixel signal words to select the desired object, here triangle **14**, from the field for further processing as hereinafter

described. One suitable means to distinguish the triangle from other objects in the field includes means for comparing the matrix of pixel signal words to a preselected matrix of words as, for example, inputed via terminal **20**. Through terminal **20** the system operator inputs data into CPU **18** to initialize the same with a initial base matrix of signal words for comparison to the matrix corresponding to the field as perceived by camera **16**. This comparison could be through intensity, shape of form, location within the field or some or all of the foregoing. The operator may pre-program the CPU with an appropriate algorithm to enable the CPU to recognize, when compared to the matrix of signal words in conjunction with camera **16**, those words of the matrix having an intensity or range of intensities or corresponding to a particular shape or positioned at a particular location within the field or any combination of the foregoing so as to distinguish the triangle from the circle and square or any other objects lying within the field as perceived by the camera. Alternatively the operator would pre-initialize the system by viewing the matrix of pixel words at the terminal as the field is initially perceived by the camera and then, through the keyboard or by a touch screen, initialize the CPU to initially recognize the location and shape and/or intensity so as to distinguish the triangle from the other objects of the field. If the selected image, here triangle **14**, is subject to change in position or shape or form other reference data may be imputed to the CPU so that the other positions or forms or ranges of positions or forms of the selected image may be compared to the matrix of pixel signal words corresponding to the field as perceived by camera **16** for selection of the desired image from the field.

Again recognition based upon intensity may play a part in the identification and selection of the desired image.

When initialized (or pre-initialized) CPU would be pre-programmed not only as to form or location but also as to the intensities relating to the selected object.

Therefore, CPU operates upon the matrix of pixel signal words to compare and distinguish the selected image triangle **14** from the field of objects perceived by the camera. CPU then identifies the locations of those pixels which represent or correspond to the triangle and stores the locations of those pixels in memory.

To assist in identifying, distinguishing and selecting the desired object from the field, an interactive image recognition system may be provided. For example, the triangle may be provided with an infrared emitter or reflector which interacts with a receiver. Receiver perceives the field in a known relationship to the field as perceived by the camera and identifies the location of the emitted or reflected light and provides that location, for example an X_n, y_m position to the CPU. A related interactive location identification system is described in U.S. Pat. No. 3,798,795 issued on Mar. 26, 1974 to Michelsen the disclosure of which is hereby incorporated by reference. CPU then compares the position as determined by the interactive means described above to the location of the pixels locations corresponding to the selected image to assist in identifying and recognizing the selected image. Alternatively the selected image may be provided with a device to emit or reflect a certain spectra of electromagnetic radiation, eg. light of a specific wavelength as emitted or reflected by a laser. A receptor identifies, as above, the location of the source of the emitted or reflected electromagnetic radiation, and provide that matrix location to the CPU.

Also included, according to the present invention, is an array **22** of electro-optical devices transformable between a first condition and a second condition in response to electrical stimulation. For example array **22** may be a liquid crystal device or LCD including an array of devices each transformable, in response to electronic stimulus, between a condition of transparency and opaqueness. Driver **24** drives each device in the LCD array between the aforementioned conditions in a manner well known according to the prior art.

Driver **24** communicates with image processor **26** which may be part of but in any event communicates with the CPU. Image processor **26** is initialized by the CPU to drive driver **24** to render those devices in the array corresponding to a facsimile or silhouette of the selected image, triangle **14**, transparent whereas the remainder of the devices are opaque as is shown in FIG. **1**. The initial drive signals are stored in a suitable memory associated with the image processor.

As each frame of the perception of the field of images is received by the camera, the selected image (triangle) is identified and distinguished from the field of images.

CPU segregates the locations of the pixel signal words representative of the triangle and issues those signals to the image processor. The image processor compares those signals to those stored in its memory. If a change in the form or location of the selected object is noted, the memory is updated. Further the driver is driven to make the appropriate changes in the LCD to continue to generate a transparent facsimile of and corresponding to the selected image. If there is no change in form or location, the driver continues to drive the LCD as before. As can be appreciated, therefore, LCD is driven to generate a facsimile of the selected object on a continuous basis so as to mirror any changes in form or location of the image.

Turning to FIG. **2** the LCD array is seen incorporated into light **28**. Light **28** includes a housing **30** having a chimney **32** therethrough. Chimney **32** is defined by transparent walls **34** one of which mounts the LCD. Cooling air, whether by natural convection of by forced draft by employment of a fan passes through chimney to cool the LCD. A lamp **36** generates incoherent light which is focused by a concave mirror **38** to pass through chimney and LCD as shown in the drawings. Also provided are lenses **40, 42** which act in a well known fashion to focus the light having passed through the LCD. As hereinafter described lenses **40,42** may be mounted to be driven by suitable devices along their optical axes for remote focus. As can be appreciated light from the lamp passes through the LCD and is focused by the lenses to illuminate any objects in its path.

When the method and devices according to the present invention are operated, the LCD array is updated and driven to generate a transparent facsimile of the triangle thereat to the exclusion of the remainder of the field.

Light from the lamp is cast through the LCD array and is focused to illuminate only the triangle to the exclusion of the other objects of the field. Though the triangle may change in form or location, this change is noted and the LCD array is appropriately updated to continue to generate a corresponding facsimile for illumination of only the triangle.

Should the selected image be not only changeable in form and location but also be able C-o traverse an area which would fall outside of the field of view of fixed camera **18**, suitable tracking means are provided to track and maintain the selected object in the field of view of the camera. Accordingly the camera and light are mounted together for articulation about x, y, and z axes as is illustrated in FIG. **1**, as are known in the prior art, are provided to manipulate articulation of the camera/light unit. The camera and LCD and the lenses of the light may be selected or arranged such the field of view of the camera corresponds to the field of full illumination of the light as cast and focused by the light.

As described above, the field is perceived by the camera and a matrix of pixel signal words are generated. CPU, by the methods described above, recognizes and selects the selected object (triangle) CPU compares the location of the pixels corresponding to the selected image as perceived by the camera to those stored in the memory which correspond to the location of the selected image pixels from the prior frame. Should the location comparison indicate that the selected object is nearing the limits or bounds of the matrix, the camera/light unit is articulated to return or center the selected object within the field of perception of the camera and within the corresponding pixel matrix. As each frame of signals are received from the camera, CPU compares and updates thereby providing feedback as to the orientation of the selected image within the field of perception. Of course there may be manual control over the camera/light unit via a joystick or the like. The interactive recognition system described above may also be used to assist in tracking of the selected image.

Throughout the foregoing procedure, the LCD array is driven to illuminate the selected object as described above.

Focusing may also be provided. In that the camera and light are married to cooperate in the manner described, mutual focusing may be provided as a single focus drive unit **46**.

The intensity of the words of the pixel signal array corresponding to the selected image are continuously compared to those stored in the memory of the CPU. Should the intensity at those pixels representing the selected image or those at the bounds of the selected image fall off in a manner and quantity indicating an out of focus condition, CPU drives the focus controller **46** to sharpen the image as perceived by the camera and as illuminated by the light. The CPU may be pre-programmed to assume that the selected object is approaching and to activate the focus control accordingly for a first time period. If the intensity indicates a "better" focus, focus drive is continued until the intensity, for example at the bounds of the image, corresponds to a predetermined condition of focus. If during the initial time period the out of focus condition continues, a seeking procedure may be initiated to focus the image.

Preferably the camera/light unit is positioned at a suitable distance such that focusing is not required.

As can be appreciated from the foregoing, and in a theatrical setting, an entertainer would constitute the selected image with the field comprising the stage, props and supporting cast. The camera/light unit is positioned at a remote location with the CPU and terminal at any convenient location such as a control room. In that a light operator need not be situated with the unit, the camera/light unit can be positioned atop a boom or other location without regard to operator safety. The system is the initialized as by, for example, pre-programming the CPU to recognize the entertainer and his/her form and location and light intensity at certain time intervals such as during a dress rehearsal. Of course the interactive recognition means described above may be provided on the entertainer to assist in recognition. Furthermore the system may be only initialized by establishing the form, location and reflected light intensity of the entertainer at the beginning of each act or sketch with the methods and devices according to the present invention recognizing and tracking the entertainer.

Throughout the presentation the method and devices according to the present invention operate to generate a facsimile of the entertainer and, via that facsimile, illuminate only the entertainer to the exclusion of the field.

Of course it is to be understood that the methods according to the present invention may be adapted to recognize, track, and generate facsimiles for two or more object in the field.

OTHER APPLICATIONS

The present invention has a wide variety of applications hereinafter discussed.

With reference to FIG. 3, a color camera **50** is provided which generates signals ***52a***, ***52b***, ***52c*** each corresponding, respectively, to the colors red, green and blue of the field as perceived by camera **50**.

From camera **50** corresponding red, green and blue matrices of pixel signal words are generated. Accordingly each matrix of pixel signal words represents, in intensity and location, the red, green and blue colors of the field.

In the manner described above from each of these pixel signals matrices, CPU is pre-programmed to select those matrix locations or pixels which will be issued to drive the LCD array corresponding to the appropriate color. Low intensity pixel words may be eliminated in that these would indicate background. LCD driver red **54** is then driven to generate a facsimile of the red elements of the field to the exclusion of non-red colored elements of the field.

Similarly LCD driver green **56** is driven to generate a facsimile of the green elements of the field and LCD driver blue **58** is driven to generate a facsimile of the blue elements of the field. The corresponding color of light is projected through each LCD by corresponding red, green and blue light sources **60**, **62** and **64** and its facsimile and is focused to recreate, in color, the field as perceived by the camera.

Of course it is to be understood that any particular image from the field may be selected for the aforementioned treatment to the exclusion of the other elements of the field in the manner described above.

In still another application of the present invention the facsimile created thereby may be used to provide a silhouette which can be artistically completed for each frame of animation to generate an animated character. Accordingly the facsimile as generated by the CPU is stored by any suitable means for playback for animation. By simply viewing a live person or animal with the camera and methods according to the present invention a suitable silhouette or facsimile is created.

In still another application of the method according to the present invention, the camera may be adapted to perceive the infrared spectra of a field. Alternatively the camera may be a radar receiver. Signals are generated and operated upon as above to identify a desired object(s)from the field perceived by the camera or receiver. The selected object(s)are then recreated via employment of a facsimile on a screen, board or a head's up display in the manner described above.

While these and other features and advantages are apparent, it is to be understood that the foregoing description is set forth for purposes of understanding and not by way of limitation.

What is claimed is:

1. A method for generating a facsimile of an image comprising:

generating at least one electrical signal representative of a particular shape; and

arranging said at least one signal into at least one pixel signals representative of location of at least some points of the image field;

providing an array of electro-optical devices, each transformable between a first condition and a different second condition in response to electromagnetic stimulation; and

using said pixel signals to stimulate the devices in said array to generate at said array a facsimile of said particular shape.

* * * * *

EXHIBIT D

US007161562B1

(12) **United States Patent**
Hunt

(10) **Patent No.: US 7,161,562 B1**
(45) **Date of Patent: Jan. 9, 2007**

(54) **MULTILAYER CONTROL OF GOBO SHAPE**

(75) Inventor: **Mark A. Hunt**, Derby (GB)

(73) Assignee: **Production Resource Group, L.L.C.**, New Windsor, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1083 days.

(21) Appl. No.: **09/668,824**

(22) Filed: **Sep. 22, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/155,513, filed on Sep. 22, 1999.

(51) **Int. Cl.**
G09G 3/00 (2006.01)

(52) **U.S. Cl.** **345/32**; 359/291; 362/233

(58) **Field of Classification Search** 345/31, 345/32, 55; 359/291, 418, 419; 362/268, 362/286, 233, 293, 386

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,282,121 A *	1/1994	Bornhorst et al.	....	362/294
5,502,627 A *	3/1996	Hunt et al.	....	362/286
5,828,485 A	10/1998	Hewlett		
5,988,817 A	11/1999	Mizushima et al.		
6,126,288 A	10/2000	Hewlett		
6,538,797 B1 *	3/2003	Hunt	....	359/291

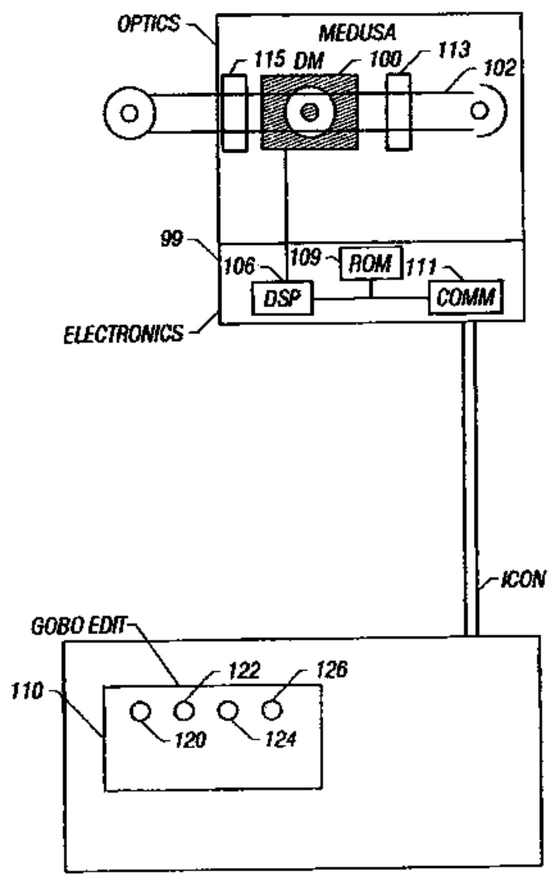
* cited by examiner

Primary Examiner—Dennis-Doon Chow
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A control of gobos defend by records in the gobo. The gobos are formed by menued shapes.

26 Claims, 11 Drawing Sheets



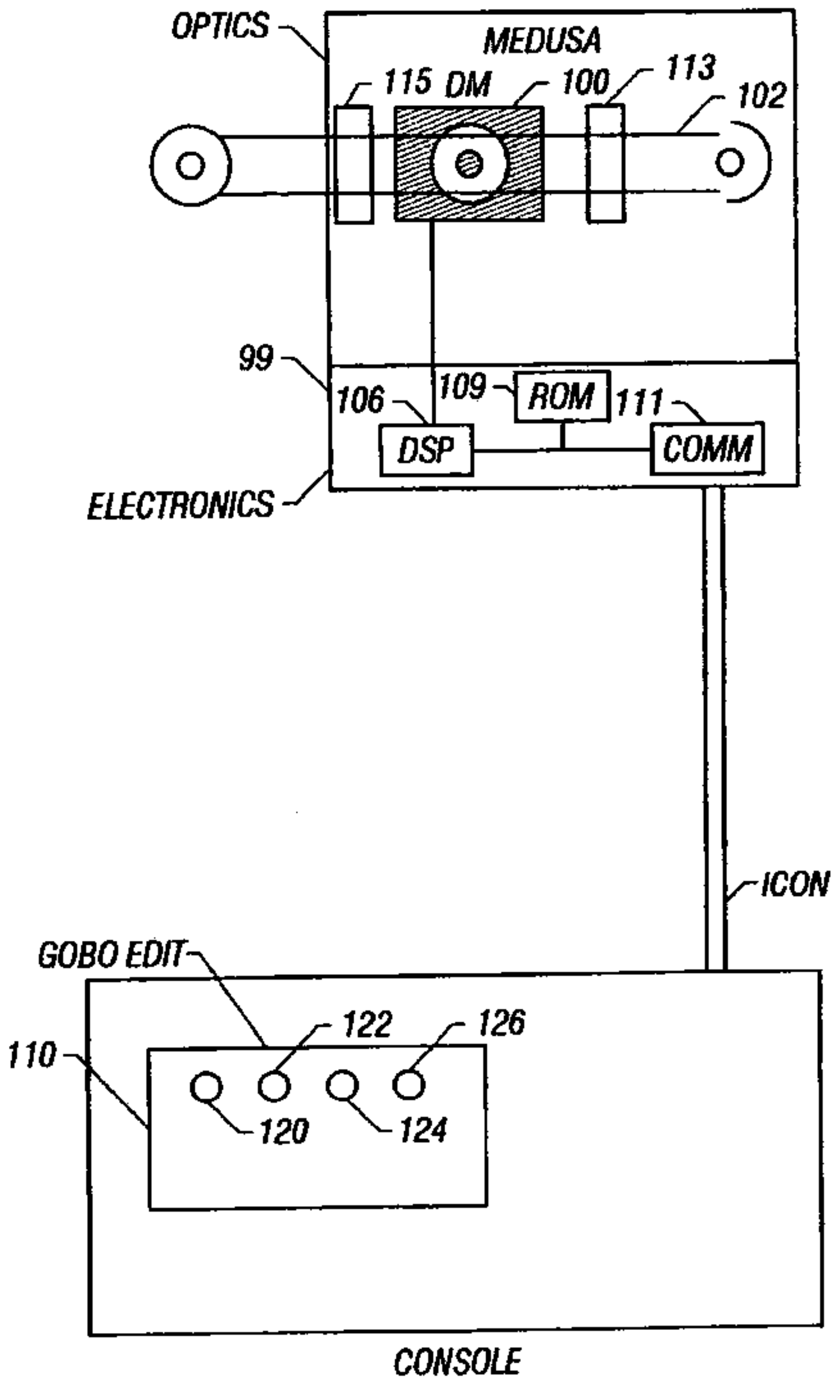

OPTICS
MEDUSA
115
DM
100
113
102
99
106
109
ROM
111
DSP
COMM
ELECTRONICS
ICON
GOBO EDIT
122
126
110
120
124
CONSOLE

FIG. 1

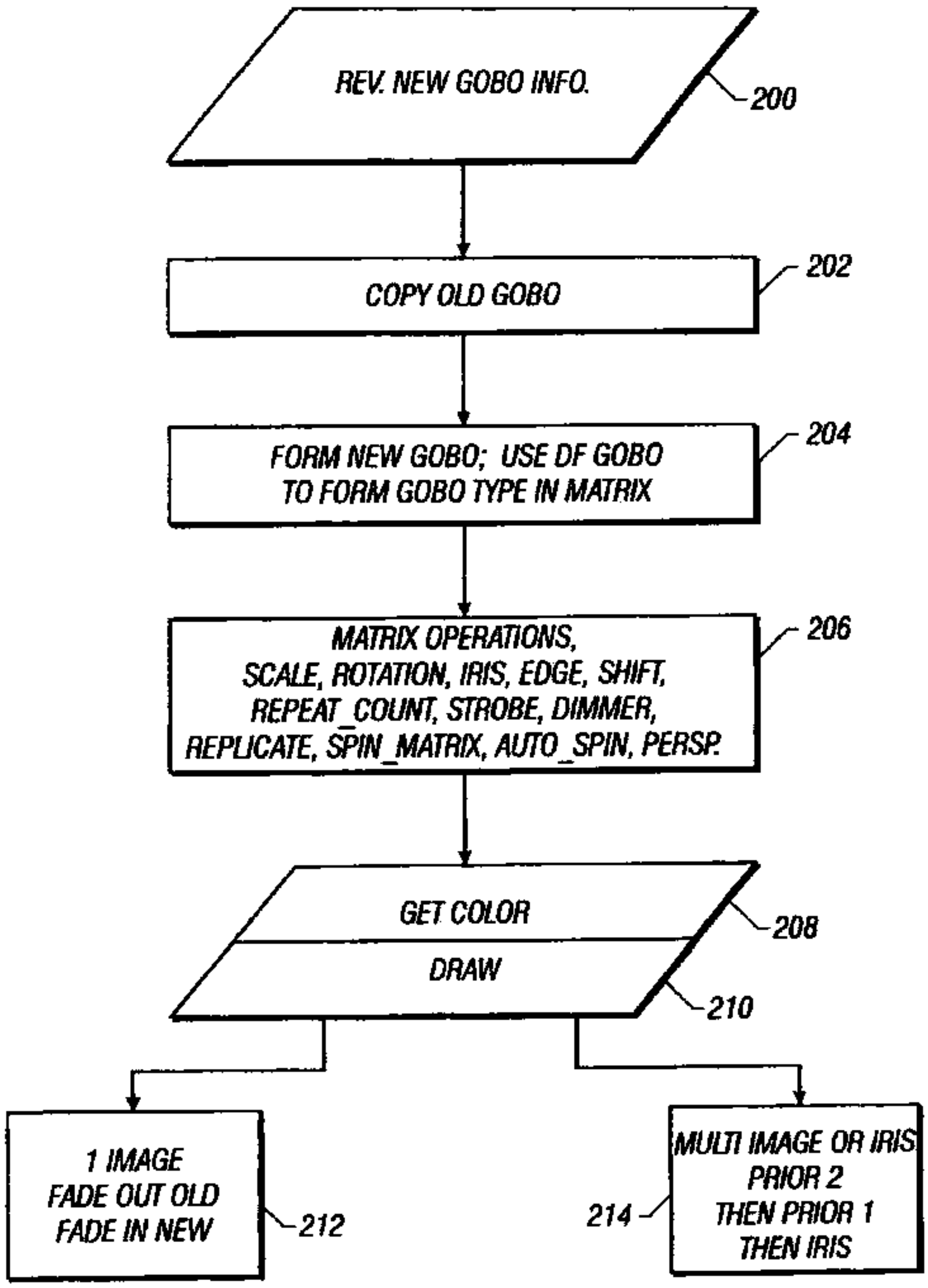

REV. NEW GOBO INFO.
200
COPY OLD GOBO
202
FORM NEW GOBO; USE DF GOBO TO FORM GOBO TYPE IN MATRIX
204
MATRIX OPERATIONS, SCALE, ROTATION, IRIS, EDGE, SHIFT, REPEAT_COUNT, STROBE, DIMMER, REPLICATE, SPIN_MATRIX, AUTO_SPIN, PERSP.
206
GET COLOR
208
DRAW
210
1 IMAGE FADE OUT OLD FADE IN NEW
212
MULTI IMAGE OR IRIS PRIOR 2 THEN PRIOR 1 THEN IRIS
214

FIG. 2

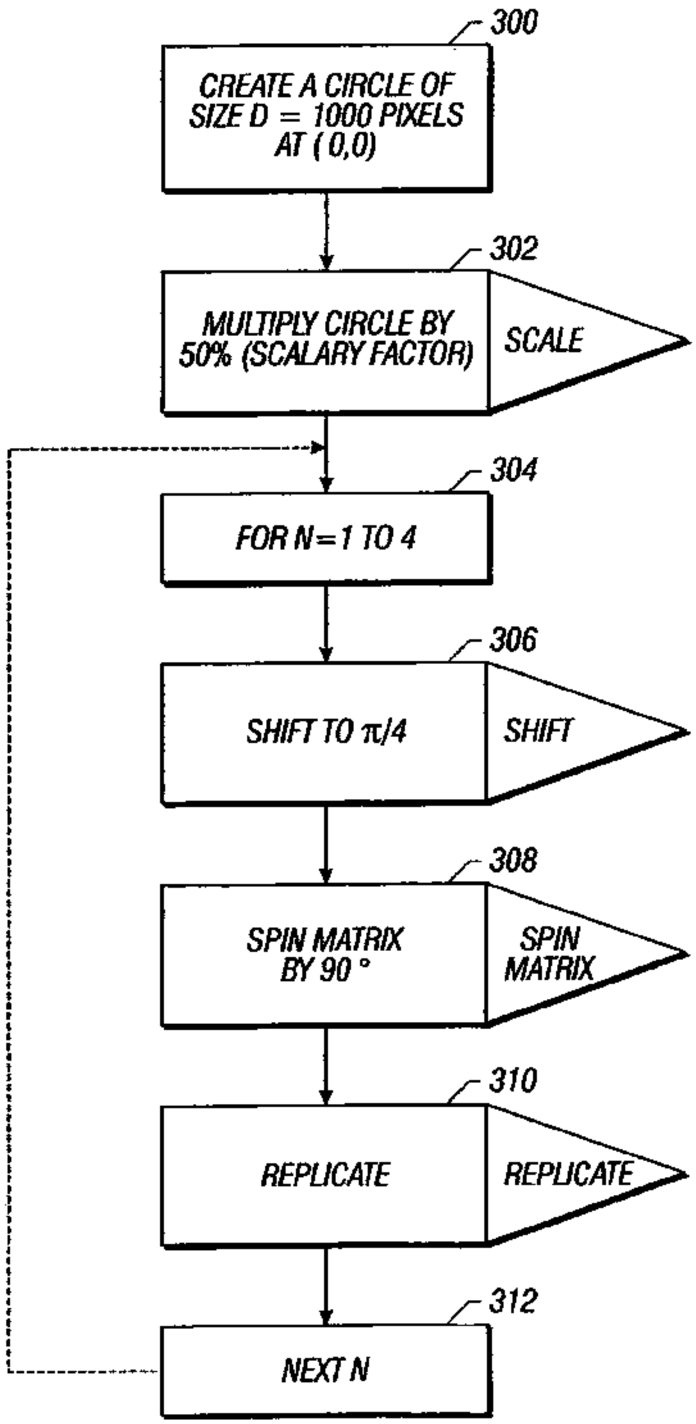

300
CREATE A CIRCLE OF SIZE D = 1000 PIXELS AT (0,0)
302
MULTIPLY CIRCLE BY 50% (SCALARY FACTOR)
SCALE
304
FOR N=1 TO 4
306
SHIFT TO π/4
SHIFT
308
SPIN MATRIX BY 90 °
SPIN MATRIX
310
REPLICATE
REPLICATE
312
NEXT N

FIG. 3

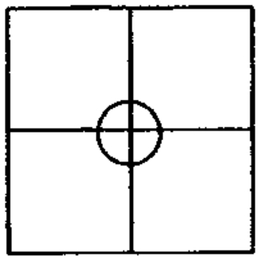


FIG. 4A

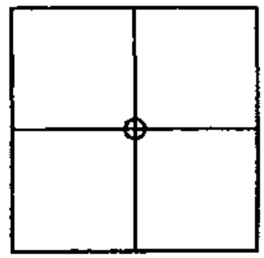


FIG. 4B

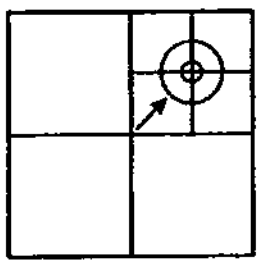


FIG. 4C

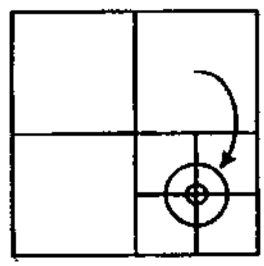


FIG. 4D

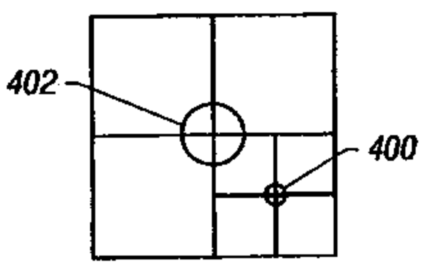


FIG. 4E

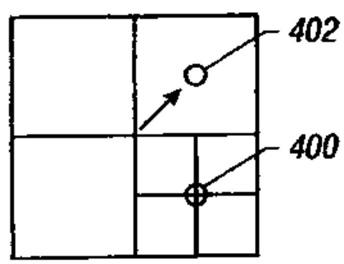


FIG. 4F

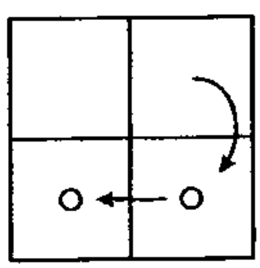


FIG. 4G

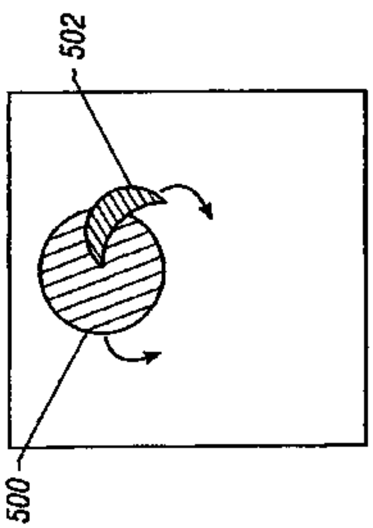


FIG. 5

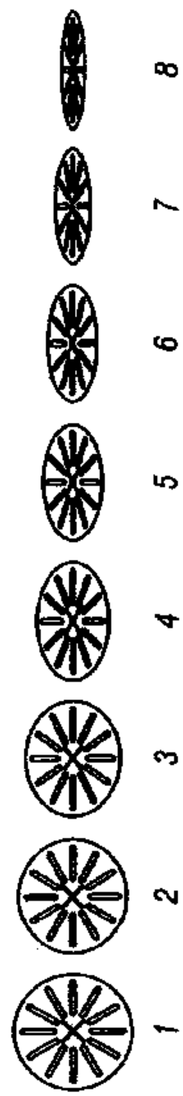


FIG. 6

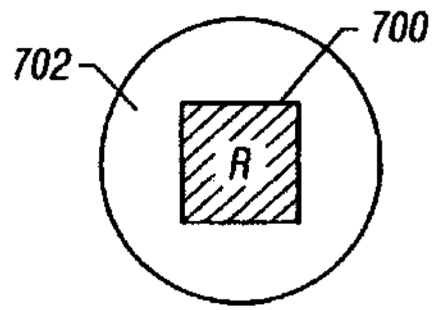


FIG. 7A

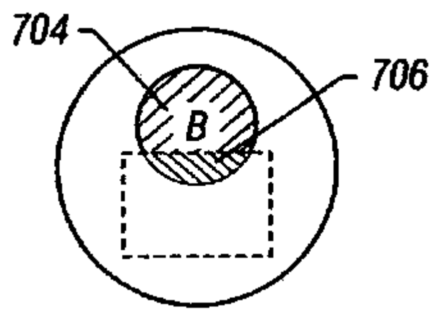


FIG. 7B

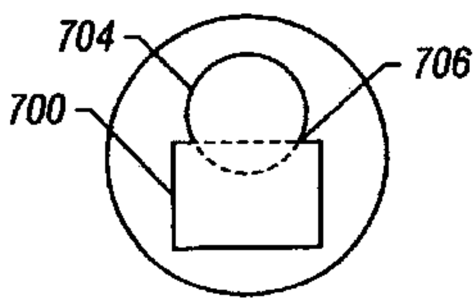
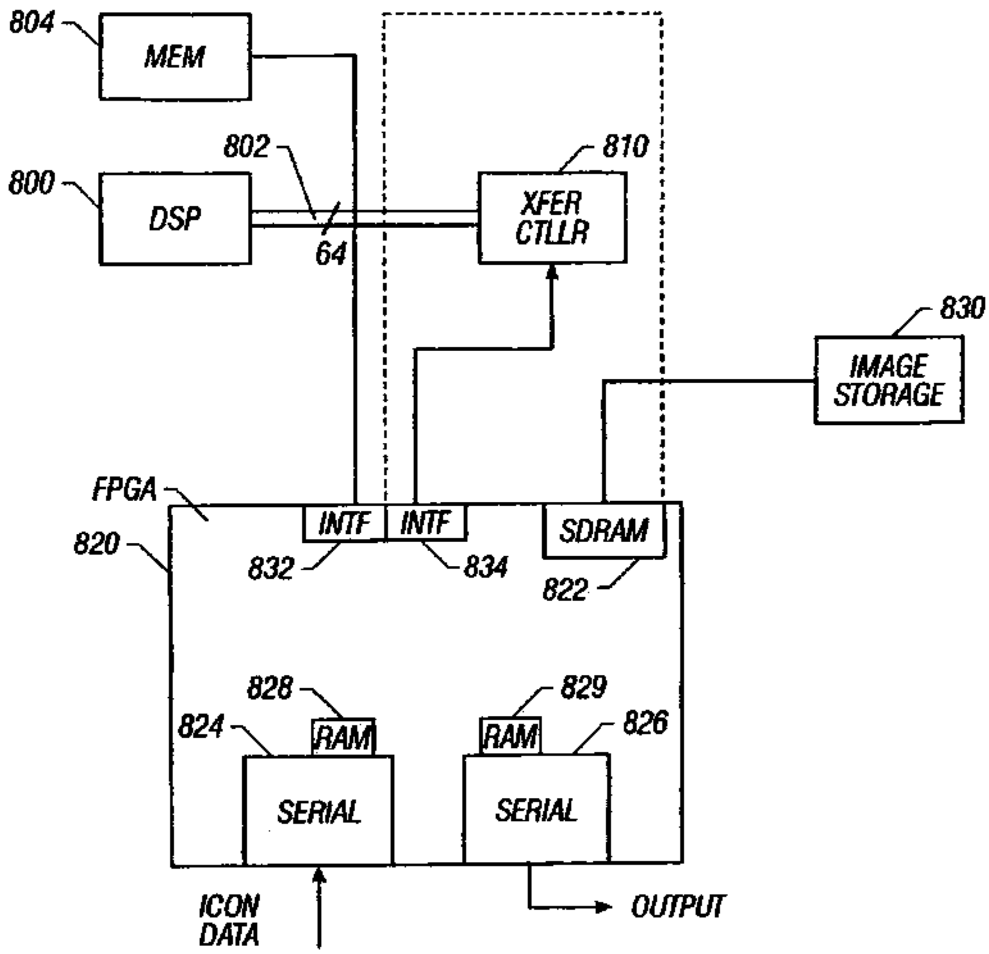
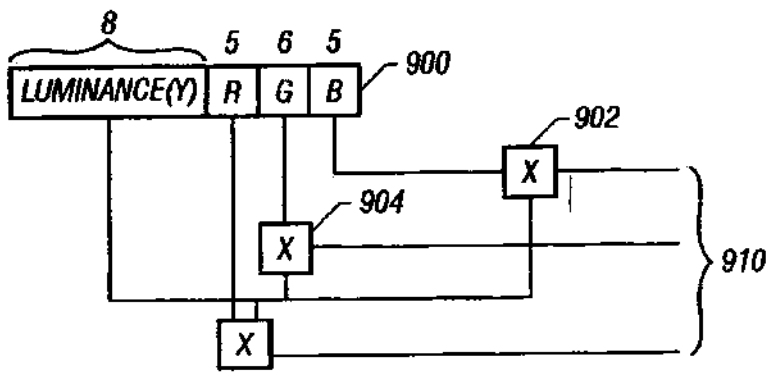


FIG. 7C



804
MEM
802
810
800
DSP
XFER CTLLR
64
830
IMAGE STORAGE
FPGA
820
INTF
INTF
SDRAM
832
834
822
828
829
824
RAM
RAM
826
SERIAL
SERIAL
ICON DATA
OUTPUT

FIG. 8



8
5
6
5
900
LUMINANCE(Y)
R
G
B
902
X
904
X
910
X

FIG. 9

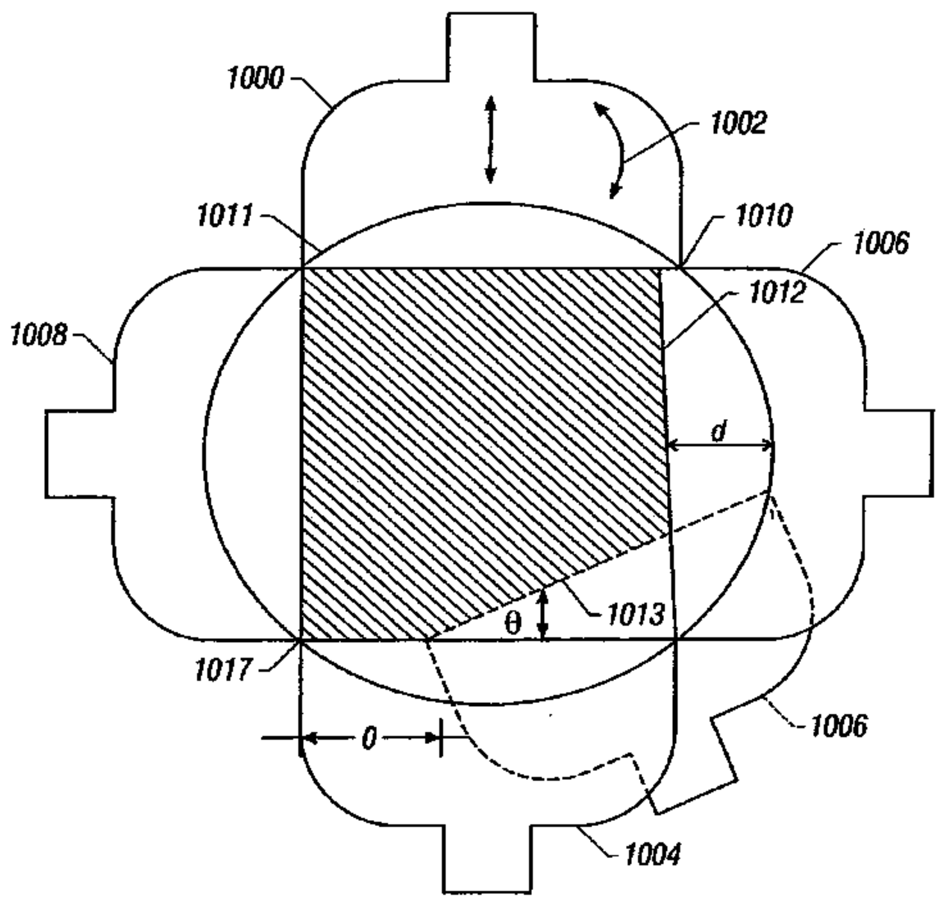


FIG. 10

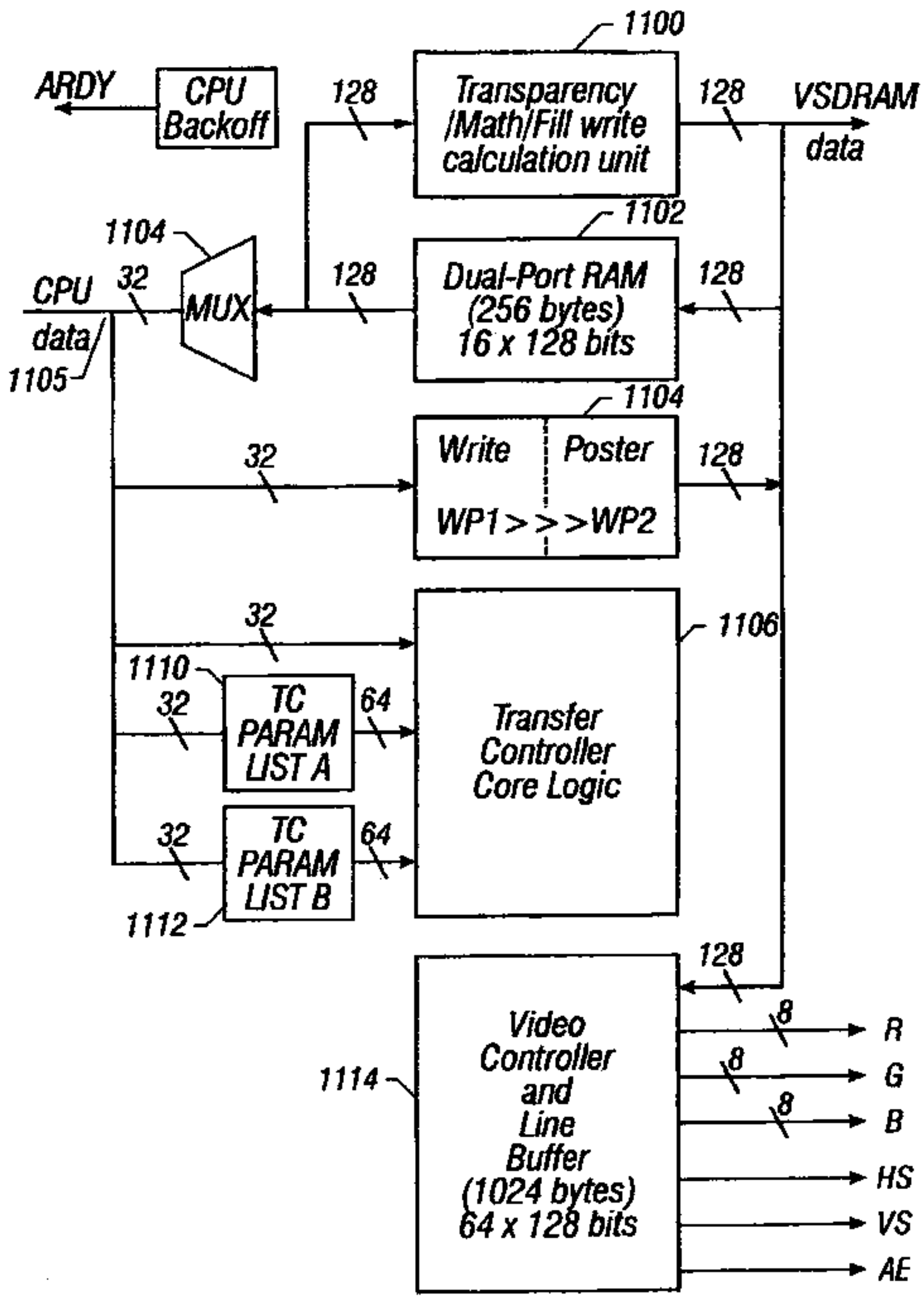

1100
ARDY
CPU Backoff
128
Transparency /Math/Fill write calculation unit
128
VSDRAM data
1102
1104
CPU data
32
MUX
128
Dual-Port RAM (256 bytes) 16 x 128 bits
128
1105
1104
32
Write
Poster
WP1>>>WP2
128
32
1106
1110
32
TC PARAM LIST A
64
Transfer Controller Core Logic
32
TC PARAM LIST B
64
1112
128
1114
Video Controller and Line Buffer (1024 bytes) 64 x 128 bits
8
R
8
G
8
B
HS
VS
AE

FIG. 11

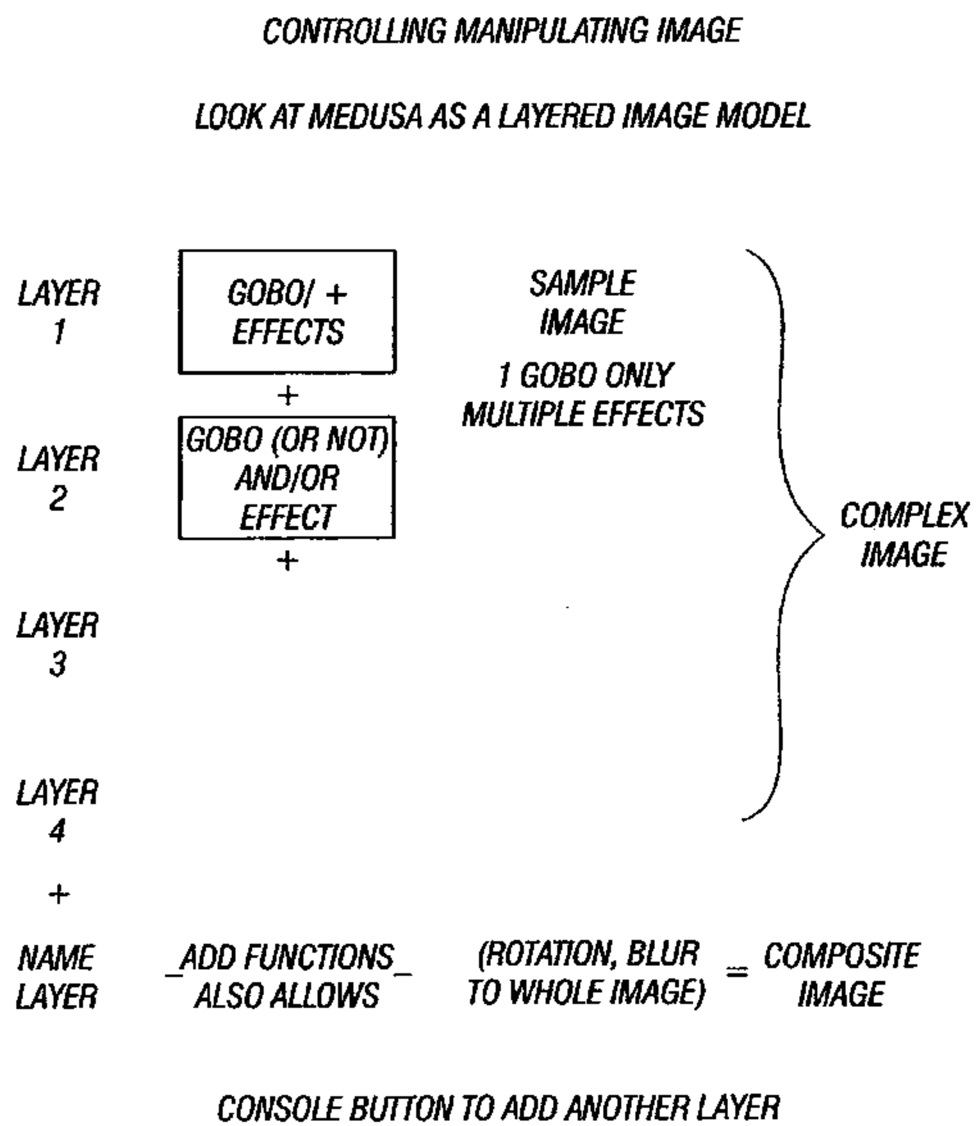

CONTROLLING MANIPULATING IMAGE
LOOK AT MEDUSA AS A LAYERED IMAGE MODEL
LAYER 1
GOBO/ + EFFECTS
+
SAMPLE IMAGE
1 GOBO ONLY MULTIPLE EFFECTS
LAYER 2
GOBO (OR NOT) AND/OR EFFECT
+
COMPLEX IMAGE
LAYER 3
LAYER 4
+
NAME LAYER
ADD FUNCTIONS ALSO ALLOWS
(ROTATION, BLUR TO WHOLE IMAGE)
=
COMPOSITE IMAGE
CONSOLE BUTTON TO ADD ANOTHER LAYER

FIG. 12

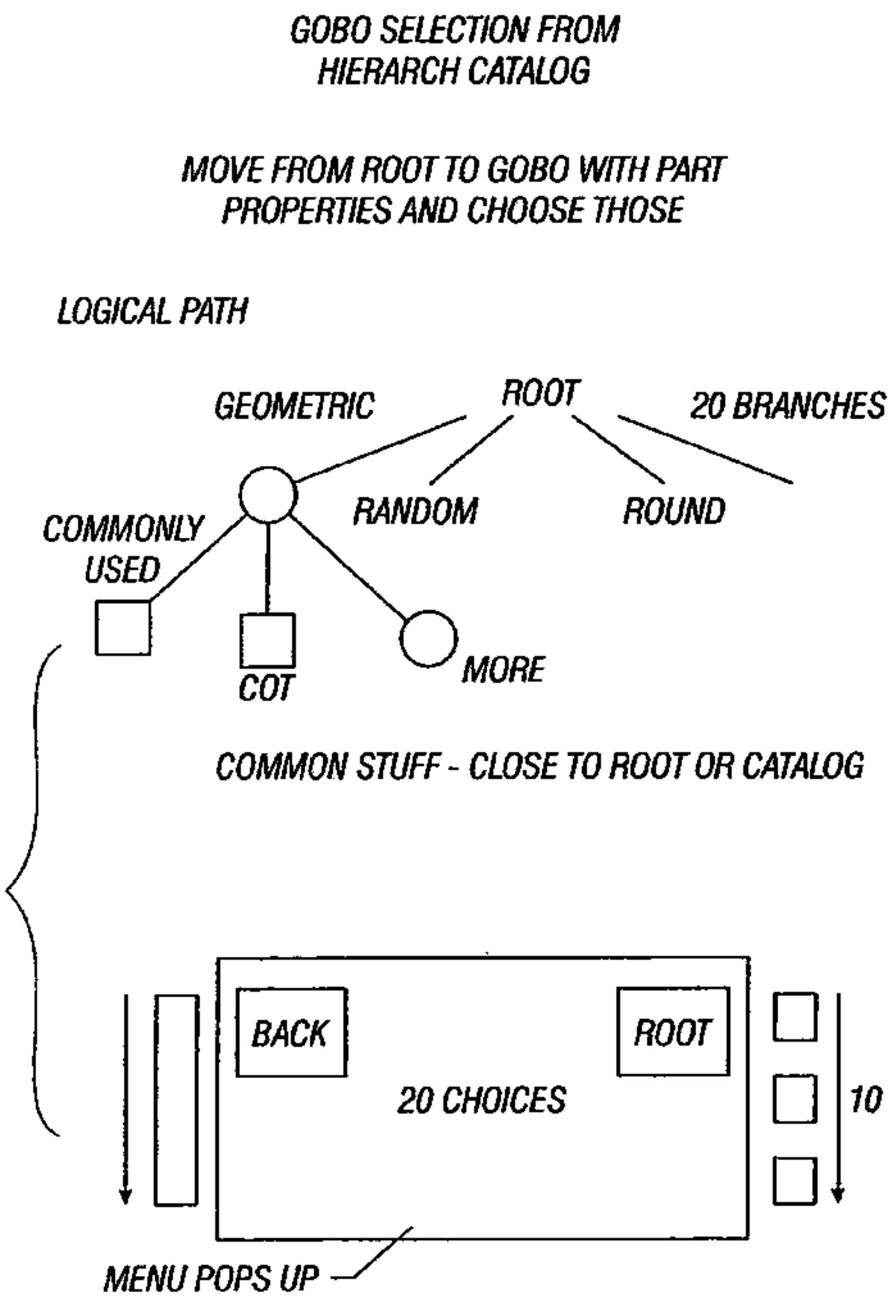

GOBO SELECTION FROM
HIERARCH CATALOG
MOVE FROM ROOT TO GOBO WITH PART
PROPERTIES AND CHOOSE THOSE
LOGICAL PATH
GEOMETRIC
ROOT
20 BRANCHES
RANDOM
ROUND
COMMONLY
USED
COT
MORE
COMMON STUFF - CLOSE TO ROOT OR CATALOG
BACK
ROOT
20 CHOICES
10
MENU POPS UP

FIG. 13

MULTILAYER CONTROL OF GOBO SHAPE

This application claims the benefit of U.S. Provisional Application No. 60/155,513, filed on Sep. 22, 1999.

FIELD

The present invention relates to a system of controlling light beam pattern ("gobo") shape in a pixilated gobo control system using a multilayer control.

BACKGROUND

U.S. Pat. No. 6,188,933, issued Feb. 13, 2001, describes a stage lighting system which operates based on computer-provided commands to form special effects. One of those effects is control of the shape of a light pattern that is transmitted by the device. This control is carried out on a pixel-by-pixel basis, hence referred to in this specification as pixilated. Control is also carried out using an x-y controllable device. The embodiment describes using a digital mirror device, but other x-y controllable devices such as a grating light valve, are also contemplated.

The computer controlled system includes a digital signal processor which is used to create an image command. That image command controls the pixels of the x-y controllable device to shape the light that it is output from the device.

The system described in the above-referenced application allows unparalleled flexibility in selection of gobo shapes and movement. This opens an entirely new science of controlling gobos.

SUMMARY

The present disclosure defines communicating with an x-y controllable device to form special electronic light pattern shapes. More specifically, the present application describes different aspects of communication with an electronic gobo. These aspects include improved processing or improved controls for the gobo and various ways of forming the user interface for such a device.

The present specification discloses controlling gobos based on layers. A complex light passing outline is defined by a number of different layers, each of which includes some number of items for the gobo.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects of the invention will now be described with reference to the attached drawings, in which:

FIG. 1 shows a block diagram of the basic system operating the embodiment;

FIG. 2 shows a basic flowchart of operation;

FIG. 3 shows a flowchart of forming a replicating circles type gobo;

FIGS. 4A through 4G show respective interim results of carrying out the replicating circles operation;

FIG. 5 shows the result of two overlapping gobos rotating in opposite directions; and

FIGS. 6(1) through 6(8) show a z-axis flipping gobo.

FIGS. 7A–7C show overlapping square gobos.

FIG. 8 shows the DSP for this operation.

FIG. 9 shows a y/c conversion.

FIG. 10 shows a framing shutter.

FIG. 11 shows a transfer controller.

FIG. 12 shows a layout of the layered system.

FIG. 13 shows a gobo selection tree.

DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 shows a block diagram of the hardware used according to the preferred embodiment. As described above, this system uses a digital mirror device 100, which has also been called a digital mirror device ("DMD") and a digital light processor device ("DLP"). More generally, any system which allows controlling shape of light on a pixel basis, including a grating light valve, could be used as the light shaper. This light shaper forms the shape of light which is transmitted. FIG. 1 shows the light being transmitted as 102, and shows the transmitted light. The information for the digital mirror 100 is calculated by a digital signal processor 106. Information is calculated based on local information stored in the lamp, e.g., in ROM 109, and also in information which is received from the console 104 over the communication link.

The operation is commanded according to a format.

The preferred data format provides 4 bytes for each of color and gobo control information.

The most significant byte of gobo control data, ("dfGobo") indicates the gobo type. Many different gobo types are possible. Once a type is defined, the gobo formed from that type is represented by a number. That type can be edited using a special gobo editor described herein. The gobo editor allows the information to be modified in new ways, and forms new kinds of images and effects.

The images which are used to form the gobos may have variable and/or moving parts. The operator can control certain aspects of these parts from the console via the gobo control information. The type of gobo controls the gobo editor to allow certain parameters to be edited.

The examples given below are only exemplary of the types of gobo shapes that can be controlled, and the controls that are possible when using those gobo shapes. Of course, other controls of other shapes are possible and predictable based on this disclosure.

First Embodiment

A first embodiment is the control of an annulus, or "ring" gobo. The DMD 100 in FIG. 1 is shown with the ring gobo being formed on the DMD. The ring gobo is type 000A. When the gobo type 0A is enabled, the gobo editor 110 on the console 104 is enabled and the existing gobo encoders 120, 122, 124, and 126 are used. The gobo editor 110 provides the operator with specialized control over the internal and the external diameters of the annulus, using separate controls in the gobo editor.

The gobo editor and control system also provides other capabilities, including the capability of timed moves between different edited parameters. For example, the ring forming the gobo could be controlled to be thicker. The operation could then effect a timed move between these "preset" ring thicknesses. Control like this cannot even be attempted with conventional fixtures.

Another embodiment is a composite gobo with moving parts. These parts can move though any path that are programmed in the gobo data itself. This is done in response to the variant fields in the gobo control record, again with timing. Multiple parts can be linked to a single control allowing almost unlimited effects.

Another embodiment of this system adapts the effect for an "eye" gobo, where the pupil of the eye changes its position (look left, look right) in response to the control.

Yet another example is a Polygon record which can be used for forming a triangle or some other polygonal shape.

The control can be likened to the slider control under a QuickTime movie window, which allows you to manually move to any point in the movie. However, our controls need not be restricted to timelines.

Even though such moving parts are used, scaling and rotation on the gobo is also possible.

The following type assignments are contemplated:

00_0F=FixedGobo (with no "moving parts")

10_1F=SingleCntrl (with 1 "moving part")

20_2F=DoubleCntrl (with 2 "moving parts")

30_FF=undefined, reserved.

The remaining control record bytes for each type are defined as follows:

Byte: memory	dfGobo2	dtGobo3	dfGobo4	#gobos/type, total
_ FixedGobo: 256M	ID[23:16]	ID[15:8]	ID[7:0]	16M/type
SingleCntrl:	ID[15:8]	ID[7:0]	control#1	64k/type
1M DoubleCntrl: 4k	ID[7:0]	control#2	control#1	256/type

As can be seen from this example, this use of the control record to carry control values does restrict the number of gobos which can be defined of that type, especially for the 2-control type.

Console Support:

The use of variant part gobos requires no modifications to existing console software for the ICON (7M) console. The Gobo editor in current ICON software already provides 4 separate encoders for each gobo. These translate directly to the values of the 4 bytes sent in the communications data packet as follows:

Byte:	dfGobo	dfGobo2	dfGobo3dfGobo4
Enc:	TopRight	MidRight	BotRightBotLeft
FixedGobo:	ID[23:16]	ID[15:8]	ID[7:0]
SingleCntrl:	ID[15:8]	ID[7 0]	control#1
DoubleCntrl:	ID[7:0]	control#2	control#1

These values would be part of a preset gobo, which could be copied as the starting point.

Once these values are set, the third and fourth channels automatically become the inner/outer radius controls. Using two radii allows the annulus to be turned "inside out".

Each control channel's data always has the same meaning within the console. The console treats these values as simply numbers that are passed on. The meanings of those numbers, as interpreted by the lamps change according to the value in dfGobo.

The lamp will always receives all 4 bytes of the gobo data in the same packet. Therefore, a "DoubleCntrl" gobo will always have the correct control values packed along with it.

Hence, the console needs no real modification. If a "soft" console is used, then name reassignments and/or key reassignments may be desirable.

Timing:

For each data packet, there is an associated "Time" for gobo response. This is conventionally taken as the time allotted to place the new gobo in the light gate. This delay has been caused by motor timing. In this system, variant gobo, the control is more dynamically used. If the non-variant parts of the gobo remain the same, then it is still the same gobo, only with control changes. Then, the time value is interpreted as the time allowed for the control change.

Since different gobo presets (in the console) can reference the same gobo, but with different control settings, this allows easily programmed timed moves between different annuli, etc.

Internal Workings:

When the gobo command data is extracted from the packet at the lamp, the dfGobo byte is inspected first, to see if either dfGobo3 or dfGobo4 are significant in selecting the image. In the case of the "Cntrl" variants, one or both of these bytes is masked out, and the resulting 32-bit number is used to search for a matching gobo image (by Gobo _ID) in the library stored in the lamp's ROM **109**.

If a matching image is found, and the image is not already in use, then the following steps are taken:

1) The image data is copied into RAM, so that its fields may be modified by the control values. This step will be skipped if the image is currently active.

2) The initial control values are then recovered from the data packet, and used to modify certain fields of the image data, according to the control records.

3) The image is drawn on the display device, using the newly-modified fields in the image data.

If the image is already in use, then the RAM copy is not altered. Instead, a time-sliced task is set up to slew from the existing control values to those in the new data packet, in a time determined by the new data packet.

At each vertical retrace of the display, new control values are computed, and steps **2** (using the new control values) and **3** above are repeated, so that the image appears modified with time.

The Image Data Records:

All images stored in the lamp are in a variant record format:

Header:

Length 32 bits, offset to next gobo in list.

Gobo _ID 32 bits, serial number of gobo.

Gobo records:

Length 32 bits, offset to next record.

Opcode 16 bits, type of object to be drawn.

Data Variant part—data describing object.

_Length 32 bits, offset to next record.

Opcode 16 bits, type of object to be drawn.

Data Variant part—data describing object.

_EndMarker 64 bits, all zeroes—indicates end of gobo data.

+Next gobo, or End Marker, indicating end of gobo list.

Gobos with controls are exactly the same, except that they contain control records, which describe how the control values are to affect the gobo data. Each control record contains the usual length and Opcode fields, and a field containing the control number (**1** or **2**).

These are followed by a list of "field modification" records. Each record contains information about the offset (from the start of the gobo data) of the field, the size (8, 16 or 32 bits) of the field, and how its value depends on the control value.

Length 32 bits, offset to next record

Opcode 16 bits=control_record (constant)

CntrlNum 16 bits=1 or 2 (control number)

/* field modification record #1 */

Address 16 bits, offset from start of gobo to affected field.

Flags 16 bits, information about field (size, signed, etc)
Scale 16 bits, scale factor applied to control before use
zPoint 16 bits, added to control value after scaling.
/* field modification record #2 */
Address 16 bits, offset from start of gobo to affected field.
Flags 16 bits, information about field (size, signed, etc)
Scale 16 bits, scale factor applied to control before use
zPoint 16 bits, added to control value after scaling.

As can be seen, a single control can have almost unlimited effects on the gobo, since ANY values in the data can be modified in any way, and the number of field modification records is almost unlimited.

Note that since the control records are part of the gobo data itself, they can have intimate knowledge of the gobo structure. This makes the hard-coding of field offsets acceptable.

In cases where the power offered by this simple structure is not sufficient, a control record could be defined which contains code to be executed by the processor. This code would be passed parameters, such as the address of the gobo data, and the value of the control being adjusted.

EXAMPLE RECORDS

The Annulus record has the following format:
Length 32 bits
Opcode 16 bits, =type_annulus
Pad 16 bits, unused
Centre_x 16 bits, x coordinate of centre
Centre_y 16 bits, y coordinate of centre
OuterRad 16 bits, outside radius (the radii get swapped when drawn if their values are in the wrong order)
InnerRad 16 bits, inside radius

It can be seen from this that it is easy to "target" one of the radius parameters from a control record. Use of two control records, each with one of the radii as a target, would provide full control aver the annulus shape.

Note that if the center point coordinates are modified, the annulus will move around the display area, independent of any other drawing elements in the same gobo's data.

The Polygon record for a triangle has this format:
Length 32 bits
Opcode 16 bits, =type_polygon
Pad 16 bits, vertex count=3
Centre_x 16 bits, x coordinate of vertex
Centre_y 16 bits, y coordinate of vertex
Centre_x16 bits, x coordinate of vertex
Centre_y 16 bits, y coordinate of vertex
Centre_x 16 bits, x coordinate of vertex
Centre_y 16 bits, y coordinate of vertex

It is easy to modify any of the vertex coordinates, producing distortion of the triangle.

The gobo data can contain commands to modify the drawing environment, by rotation, scaling, offset, and color control, the power of the control records is limitless.

Second Embodiment

This second embodiment provides further detail about implementation once the gobo information is received.

Gobo information is, at times, being continuously calculated by DSP **106**. The flowchart of FIG. **2** shows the handling operation that is carried out when new gobo information is received.

At step **200**, the system receives new gobo information. In the preferred embodiment, this is done by using a communications device **111** in the lamp **99**. The communications device is a mailbox which indicates when new mail is received. Hence, the new gobo information is received at step **200** by determining that new mail has been received.

At step **202**, the system copies the old gobo and switches pointers. The operation continues using the old gobo until the draw routine is called later on.

At step **204**, the new information is used to form a new gobo. The system uses a defined gobo ("Gobo") as discussed previously which has a defined matrix. The type dfGobo is used to read the contents from the memory **109** and thereby form a default image. That default image is formed in a matrix. For example, in the case of an annulus, a default size annulus can be formed at position 0,0 in the matrix. An example of forming filled balls is provided herein.

Step **206** represents calls to subroutines. The default gobo is in the matrix, but the power of this system is its ability to very easily change the characteristics of that default gobo. In this embodiment, the characteristics are changed by changing the characteristics of the matrix and hence, shifting that default gobo in different ways. The matrix operations, which are described in further detail herein, include scaling the gobo, rotation, iris, edge, strobe, and dimmer. Other matrix operations are possible. Each of these matrix operations takes the default gobo, and does something to it.

For example, scale changes the size of the default gobo.** Rotation rotates the default gobo by a certain amount. Iris simulates an iris operation by choosing an area of interest, typically circular, and erasing everything outside that area of interest. This is very easily done in the matrix, since it simply defines a portion in the matrix where all black is written.

Edge effects carry out certain effects on the edge such as softening the edge. This determines a predetermined thickness, which is translated to a predetermined number of pixels, and carries out a predetermined operation on the number of pixels. For example, for a 50% edge softening, every other pixel can be turned off. The strobe is in effect that allows all pixels to be turned on and off at a predetermined frequency, i.e., 3 to 10 times a second. The dimmer allows the image to be made dimmer by turning off some of the pixels at predetermined times.

The replicate command forms another default gobo, to allow two different gobos to be handled by the same record. This will be shown with reference to the exemplary third embodiment showing balls. Each of those gobos are then handled as the same unit and the entirety of the gobos can be, for example, rotated. The result of step **206** and all of these subroutines that are called is that the matrix includes information about the bits to be mapped to the digital mirror **100**.

At step **208**, the system then obtains the color of the gobos from the control record discussed previously. This gobo color is used to set the appropriate color changing circuitry **113** and **115** in the lamp **99**. Note that the color changing circuitry is shown both before and after the digital mirror **100**. It should be understood that either of those color changing circuits could be used by itself.

At step **210**, the system calls the draw routine in which the matrix is mapped to the digital mirror. This is done in different ways depending on the number of images being used. Step **212** shows the draw routine for a single image being used as the gobo. In that case, the old gobo, now copied as shown in step **202**, is faded out while the new gobo newly calculated is faded in. Pointers are again changed so

that the system points to the new gobo. Hence, this has the effect of automatically fading out the old gobo and fading in the new gobo.

Step **214** schematically shows the draw routine for a system with multiple images for an iris. In that system, one of the gobos is given priority over the other. If one is brighter than the other, then that one is automatically given priority. The one with priority **2**, the lower priority **1**, is written first. Then the higher priority gobo is written. Finally, the iris is written which is essentially drawing black around the edges of the screen defined by the iris. Note that unlike a conventional iris, this iris can take on many different shapes. The iris can take on not just a circular shape, but also an elliptical shape, a rectangular shape, or a polygonal shape. In addition, the iris can rotate when it is non-circular so that for the example of a square iris, the edges of the square can actually rotate.

Returning to step **206**, in the case of a replicate, there are multiple gobos in the matrix. This allows the option of spinning the entire matrix, shown as thin matrix.

An example will now be described with reference to the case of repeating circles. At step **200**, the new gobo information is received indicating a circle. This is followed by the other steps of **202** where the old gobo is copied, and **204** where the new gobo is formed. The specific operation forms a new gobo at step **300** by creating a circle of size diameter equals 1000 pixels at origin **00**. This default circle is automatically created. FIG. **4A** shows the default gobo which is created, a default size circle at **00**. It is assumed for purposes of this operation that all of the circles will be the same size.

At step **302**, the circle is scaled by multiplying the entire circle by an appropriate scaling factor. Here, for simplicity, we are assuming a scaling factor of 50% to create a smaller circle. The result is shown in FIG. **4B**. A gobo half the size of the gobo of FIG. **4A** is still at the origin. This is actually the scale of the subroutine as shown in the right portion of step **302**. Next, since there will be four repeated gobos in this example, a four-loop is formed to form each of the gobos at step **304**. Each of the gobos is shifted in position by calling the matrix operator shift. In this example, the gobo is shifted to a quadrant to the upper right of the origin. This position is referred to as π over 4 in the FIG. **3** flowchart and results in the gobo being shifted to the center portion of the top right quadrant as shown in FIG. **4C**. This is again easily accomplished within the matrix by moving the appropriate values. At step **308**, the matrix is spun by 90 degrees in order to put the gobo in the next quadrant as shown in FIG. **4D** in preparation for the new gobo being formed into the same quadrant. Now the system is ready for the next gobo, thereby calling the replicate command which quite easily creates another default gobo circle and scales it. The four-loop is then continued at step **312**.

The replicate process is shown in FIG. **4E** where a new gobo **402** is formed in addition to the existing gobo **400**. The system then passes again through the four-loop, with the results being shown in the following figures. In FIG. **4F**, the new gobo **402** is again moved to the upper right quadrant (step **306**). In FIG. **4G**, the matrix is again rotated to leave room for a new gobo in the upper right quadrant. This continues until the end of the four-loop. Hence, this allows each of the gobos to be formed.

Since all of this is done in matrix operation, it is easily programmable into the digital signal processor. While the above has given the example of a circle, it should be understood that this scaling and moving operation can be carried out for anything. The polygons, circles, annulus, and any other shape is easily scaled.

The same operation can be carried out with the multiple parameter gobos. For example, for the case of a ring, the variable takes the form annulus (inner R, outer R, x and y). This defines the annulus and turns of the inner radius, the outer radius, and x and y offsets from the origin. Again, as shown in step **3**, the annulus is first written into the matrix as a default size, and then appropriately scaled and shifted. In terms of the previously described control, the ring gobo has two controls: control **1** and control **2** defined the inner and outer radius.

Each of these operations is also automatically carried out by the command repeat count which allows easily forming the multiple position gobo of FIGS. **4A**–**4G**. The variable auto spin defines a continuous spin operation. The spin operation commands the digital signal processor to continuously spin the entire matrix by a certain amount each time.

One particularly interesting feature available from the digital mirror device is the ability to use multiple gobos which can operate totally separately from one another raises the ability to have different gobos spinning in different directions. When the gobos overlap, the processor can also calculate relative brightness of the two gobos. In addition, one gobo can be brighter than the other. This raises the possibility of a system such as shown in FIG. **5**. Two gobos are shown spinning in opposite directions: the circle gobo **500** is spinning the counterclockwise direction, while the half moon gobo **502** is spinning in the clockwise direction. At the overlap, the half moon gobo which is brighter than the circle gobo, is visible over the circle gobo. Such effects were simply not possible with previous systems. Any matrix operation is possible, and only a few of those matrix operations have been described herein.

A final matrix operation to be described is the perspective transformation. This defines rotation of the gobo in the Z axis and hence allows adding depth and perspective to the gobo. For each gobo for which rotation is desired, a calculation is preferably made in advance as to what the gobo will look like during the Z axis transformation. For example, when the gobo is flipping in the Z axis, the top goes back and looks smaller while the front comes forward and looks larger. FIGS. **5A**–**5C** show the varying stages of the gobo flipping. In FIG. **5D**, the gobo has its edge toward the user. This is shown in FIG. **5D** as a very thin line, e.g., three pixels wide, although the gobo could be zero thickness at this point. Automatic algorithms are available for such Z axis transformation, or alternatively a specific Z axis transformation can be drawn and digitized automatically to enable a custom look.

Third Embodiment

The gobo record format described above can have two gobos therein. These two gobos can be gobo planes, which can be used to project one image superimposed over another image in a predefined way. For example, a first image can be a pattern that emits light, e.g., a standard gobo. The second image can be totally transparent, or can have holes through which the first image can be seen.

Analog gobos often project light through two gobos. The light is then projected through the intersection between the two gobos. Effectively, this takes an AND function between the gobos. Light will only be passed in places where both gobos are open.

In the present system, any function between two images can be projected as an overall gobo shape. The system can,

e.g., project the "or" between the two images. Moreover, the two images can be projected in separate colors. Therefore, for example, the system used in FIGS. 7A–7B could be carried out in software.

A first gobo shown in FIG. 7A is a square gobo. For purposes of this example, the square gobo is projected in red, forming a first lighted portion. The exterior non-projected portion **702** is black. FIG. 7B shows the second gobo to be added to the first gobo. The second gobo is an off-center circle **704** to be projected in blue. The AND between these two gobos would transmit only the intersection between the two gobos, shown by the hatched portion **706**. Moreover, this portion could only be transmitted in the additive or subtractive combination between the two colors, red and blue.

The present system defines the two images as separate planes. This enables transmitting the "or" between the two images. Therefore, both the first image **700** and the second image **704** are transmitted. Moreover, the intersection portion of the image **706** can be made in any desired color, either the color of either, the color of the subtractive combination, or a totally different color while this system describes an "or" operation, it also encompasses any combination between the gobos: e.g., X or, Schmitt-triggered (hysteresis-induced) and/or, others.

The gobo operation is simplified and made more efficient by using a transfer controller as described herein.

FIG. **8** shows the basic block diagram of this embodiment. The Digital Signal Processor (DSP) **800** effectively functions as the central processing unit. The preferred DSP for this embodiment is the TI TMS 320C80. This includes a 64-bit bus **802**. Memory **804** is attached to the bus **802**. The memory **804** effectively forms a working portion. A transfer controller **810** is provided and allows increased speed. The transfer controller can take control of the bus and can carry out certain functions. One such function is a direct memory access. This allows moving information from the program memory **804** to a desired location. The transfer controller receives information about the data to be moved, including the start location of the data, the number of bytes of the date, and the end location of the data. The destination and operation is also specified by the data **809**. The transfer controller **810** then takes the data directly from the memory **804**, processes it, and returns it to the memory or to the DMP without DSP intervention. The CPU can then therefore tell the transfer controller to take some action and then can itself do something else.

Also on the bus **802** is a hardware block **820** which is preferably formed from a Field Programmable Gate Array (FPGA). The FPGA can be configured into logical blocks as shown. The DSP also sends commands that **807** reconfigure the FPGA as needed. The FPGA can be reconfigured to form fast Synchronous Dynamic Random Access Memory (SDRAM) shown as **822**.

The preferred DSP **800** is a TI TMS 320C80. This device includes an associated transfer controller which is a combined memory controller and DMA (direct memory access) machine. It handles the movement of data and instructions within the system as required by the master processor, parallel processors, video controller, and external devices.

The transfer controller performs the following data-movement and memory-control functions:

MP and ADSP instruction-cache fills
MP data-cache fills and dirty-block write-back
MP and ADSP packet transfers (PTs)
Externally initiated packet transfers (XPTs)
VC packet transfers (VCPTs)
MP and ADSP direct external accesses (DEAs)
VC shift-register-transfer (SRTs)
DRAM refresh
sExternal bus requests

Operations are performed on the cache sub-block as requested by the processors' internal cache controllers. DEA operations transfer off-chip data directly to or from processor registers. Packet transfers are the main data transfer operations and provide an extremely flexible method for moving multidimensional blocks of data (packets) between on-chip and/or off-chip memory.

Key features of the this specific transfer controller include:

Crossbar interface
64-bit data path
Single-cycle access
External memory interface
4G-byte address range dynamically configurable memory cycles
Bus size of 8, 16, 32, or 64 bits
Selectable memory page size
Selectable row/column address multiplexing
Selectable cycle timing
Big or little indian operation Cache, VRAM, and refresh controller
Programmable refresh rate
VRAM block-write support
Independent source and destination addressing
Autonomous address generation based on packet transfer parameters
Data can be read and written at different rates
Numerous data merging and spreading functions can be performed during transfers
Intelligent request prioritization

Hence, the transfer controller allows definition of the limits of the message/data, and then the information can be automatically handled. The transfer controller also can generate a table of end points, carry out direct-memory access, and manipulate the data while transferring the data.

The SDRAM **822** can be used as fast-image memory, and can be connected, for example, to an image storage memory **830**. The FPGA can also be configured to include serial interfaces **824**, **826** with their associated RAM **828**, **829** respectively. Other hardware components can also be configured by the FPGA.

Since the FPGA can be reconfigured under control of the processor **800**, the FPGA can be reconfigured dynamically to set an appropriate amount of SDRAM **822**. For example, if a larger image or image processing area is necessary, the FPGA can be reconfigured to make more of it into image memory. If a smaller image is desired, less of the FPGA can be made into SDRAM, allowing more of the FGPA for other hardware functions. Moreover, the interfaces **832**, **834** can be dynamically reconfigured. For example, the baud rate can be changed, bus width can be reconfigured, and the like.

The serial receiver **824** receives the ICON data from the controller, as described in our copending application, 7319/63. The serial driver **826** produces a serial output that can drive, for example, an RS422 bus that runs the motors.

The C80 DSP includes the transfer controller as a part thereof. An alternative embodiment uses a different DSP. The functions of the transfer controller are then replicated in the FPGA, as desired. For example, an alternative possible DSP is the C6201 which uses the Very Large Instruction Word "VLIW" architecture. This system can use, for example, 128-bit instructions. However, since this is con-

nected to the 32-bit data bus, a transfer controller could be highly advantageous. This would enable the equivalent of direct memory access from the memory. FIG. 11 shows the gate array schematic of this alternate embodiment in which the transfer controller is part of the FPGA.

A second embodiment of the gate array logic, as preferably used according to the present system, is shown in FIG. **11**. This gate array logic is formed in the field-programmable gate array and carries out many of the functions described herein. Block **1100** corresponds to a transparency device which calculates values associated with transparency. Block **1102** is a dual-port RAM which receives the VLIW at one port thereof, and outputs that value to a multiplexer **1104**, which converts it to the 32 bits used by the CPU/DSP. The write poster **1104**. Transfer controller **1106** has the functionality discussed above, and is controlled directly by the CPU data received on line **1105**. The transfer controller can have two lists of parameters, each 64 bits in width. These values are received on the list receivers **1110**, **1112**.

Another issue noted by the current inventors is the size of images. If possible, it is desirable to avoid using uncompressed images. For example, one easy form image to manipulate is a bitmap, also known as a "*.bmp" type image. The bitmap represents each pixel of the image by a number of bits, e.g., for an 8-bit 3-primary color image, each pixel would require 24 bits. This can, unfortunately, use incredible amounts of storage. However, since the bit map has a 1-to-1 correspondence with the image, it can be relatively easy to manipulate the bit map. For example, a matrix representing the bitmap can be easily manipulated, e.g., rotated. The image form can be compressed, e.g., to a GIF or JPEG image. This image, however, loses the one-to-one correspondence and hence cannot be directly processed as easily.

One aspect of the present system is to store the image as a compressed image, and most preferably as polygons. The existing software package, Adobe Streamline (TM), breaks a bitmap into multiple polygons. The polygons can then be defined as vectors. An additional advantage is that the vectors can be easily processed by the DSP. The DSP **800** then builds the image from the vectors. Since the image is defined as vectors, it can be easily related via matrix arithmetic. Using Adobe Streamline, for example, an 800 kilobyte bit map can be compressed to a 30 kilobyte vector image.

Another improvement of the present system is the control of the gobo using filters.

In an analog gobo system, a filter can be used to blur the image, for example. Many different kinds of filters are used. For example, some filters randomly distort the image. Other filters affect the image in different ways. The blurring can be carried out as an electronic filter. A preferred user interface defines the filter as a separate gobo that is multiplied, e.g., and ed OR ed with the first gobo.

More generally, a filter can be used to alter the image in some way, e.g., scaler the image, decay the image, or the like. The blur can be used to make the image apparently out of focus in some locations. The filter uses a second gobo that simulates the effect of an analog filter. For example, one operation simulates the optical effect of the glass that forms the filter in an analog gobo. That glass is used to make an algorithm that emulates the optical properties of the glass. Those optical properties are then pushed through the matrix representing the gobo, thereby effecting a digital representation of the filter. In one aspect, the filter is considered as a separate gobo which is OR ed with the second gobo. In this case, the dual gobo definition described above can be used. Alternatively, the filter can simply be added to the gobo-defining matrix.

This definition has the advantage that it avoids defining a totally separate control. The filters are each defined as one specific gobo. There is already a manual that defines gobos. This manual has filters added to it. This avoids the need for a manual of filters.

Another aspect defined by the present system is gobos that load and execute code. Some images cannot be described in terms of control. For example, images may be defined as some random input. Some images progress with time and maintain no record of their previous state. These images are easily defined in terms of code and in terms of a progression from one time to another. Hence, the gobos that load and execute code define a gobo that includes an associated area to hold static values. A gobo is requested and the code and variables that are associated with that gobo are copied into RAM. The variables are initially at a preset state. The code that is in the gobo portion is executed, using the portions in the variables. The variables are modified at each pass through the portion.

Yet another feature of this system is intensity control over aspects of the image defining the gobo and dimming of the image defined thereby. Returning to the example of a bit map with 24-bit color, such a system would include 8 bits of red, 8 bits of green, and 8 bits of blue. It can be desirable to fade the image awhile keeping the color constant with intensity change.

One system uses an experimental technique to determine how to fade in order to maintain color constant and forms a look-up table between the constant color and the look up table.

Another system directly maps the bits to color by defining the map as chromium using techniques from color television. For example, this takes the bits, and converts the valves indicating image to color or chrominance© and image luminance (Y) of the image. The conversion between RGB and Y/C is well known. The values of Y and C which correspond to the chrominance and luminance are then stored. The gobo can then be dimmed by reducing the Y, keeping C the same. If desired, the Y/C can be converted back to RGB after dimming.

Another system allows reducing the number of bits for a bit map. Say, as an example, that it is desired to use a total of 8 bits to represent each pixel of the image. This could then be apportioned between the desired bits with red having 3 bits, green having 3 bits, and blue having 2 bits. This limits the amount of information in any of these colors. Since there are only 2 bits for blue, there are only four levels of blue that can be selected. This is often insufficient.

In this system, therefore, the bits are compressed by assuming that two adjacent lines have exactly the same values. Hence, each two lines get the same color value (but can have different intensity values). Now in a system as described above, two lines of red can have 5 bits, two lines of green can have 6 bits, and two lines of blue can also have 5 bits. This provides an appropriate dynamic range for color at the expense of losing half the resolution for color.

Moreover, this has an additional advantage in that it allows 5 bits for grey scale in such a system.

A possible problem with such a system, however, as described above, is that the information would not necessarily be aligned on byte boundaries. It could, therefore, be necessary to take the whole image, manipulate it, and then put the whole image back.

The basic system is shown in FIG. 9. The luminance Y is an 8-bit representation of the brightness level of the image. The hue is then divided into dual-line multiple bits. Each bit is used for two lines each.

Dimming in such a system is carried out as shown in FIG. 9. For example, the blue bits **900** are multiplied in a hardware multiplier **902** by the luminance. Similarly, the green is multiplied in a second hardware multiplier **904** by the same luminance value. This controls the relative levels of red, green, and blue that are output on the RGB lines **910**.

The multipliers that are used are very simple, since they simply multiply 8 bits by 3 bits. Therefore, a simple hardware multiplier can be used for this function.

This provides red, green, and blue color without loss of data and with substantially perfect fading.

An additional feature described herein is a framing shutter gobo. A basic framing shutter is shown in FIG. **10**. FIG. **10** shows the circular spot of the beam, and the analog shutter, often called a LECO. Each analog shutter **1000** can be moved in and out in the direction of the arrows shown. Each shutter can also be moved in an angular direction, shown by the arrow **1002**. There are a total of four shutters, which, in combination, enable framing the beam to a desired shape. For example, the shutter **1004** can be moved to the position shown in dotted lines as **1006**. When this happens, the effective image that is passed becomes as shown in hatched lines in FIG. **10**. Another possibility is that the shutter can be tilted to put a notch into the image.

According to this system, another record is formed for a gobo defining a framing shutter. The framing shutter gobo allows control of multiple values including the positions of the four framing shutters **1000, 1004, 1006,** and **1008**. Each framing shutter is defined in terms of its value d, corresponding to the distance between one edge **1010** of the framing shutter and the edge **1011** of the original spot. In this system, the value d is shown representing the right-hand edge of the framing shutter. Another selectable value is θ, which defines the angle that the front blade **1013** of the framing shutter makes relative to perfect horizontal or vertical. Yet another parameter which can be selected is offset O which represents the distance between the framing shutter edge **1010** and the ideal edge portion **1017**. Other values can alternatively be specified. By controlling all these values, the Medusa shutter can in effect simulate any desired framing shutter gobo.

The video controller and line buffer **1114** can also be formed from the field-programmable gate array.

A number of different special gobos are defined according to the present system. Each of these gobos is defined according to the record format described above.

These include:

Oscilloscope. This enables simulating the output value of an oscilloscope as the gobo. For example, any value that can be displayed on the oscilloscope could be used as a gobo with a finite width. This could include sine waves, square waves, straight waves, sawtooth waves, and the like.

Other variable gobos include vertical lines, moire lines, laser dots, radial lines, concentric circles, geometric spiral, bar code, moon phases, flowers and rotating flowers, a diamond tiling within a shape, kaleidoscope, tunnel vision, and others.

Animated gobos correspond to those which execute codes described above. Some examples of these include, for example, self-animating random clouds; self-animating random reflections; self-animating random flames, fireworks; randomly moving shapes such as honeycombs, crosswords, or undulations; foam; random flying shapes.

Control as a Multi-Layered Image

As described above, the present system allows sophisticated control of electronic images, and manipulation of images. The control of the gobo can be selected as a layered image mode. Each gobo then effectively becomes a multi-part layered image.

A layout of the control for the system is shown in FIG. **12**.

Layer 1 includes gobos plus effects. This is for a simple image; with one gobo only and multiple effects. The second layer includes additional information. This can include gobos, non gobos, and/or additional effects. Third layers and fourth layers are similarly situated. The multiple layers collectively form a complex image.

Effectively, therefore, a complex image is formed by a number of layers. A first layer has a gobo. Additional functions are defined in the other layers. This forms a composite image. Each layer can be operated on by the other layers. For example, each layer can be individually rotated or blurred, and/or rotation and blur can be applied to the entire image. This enables the console to be controlled incrementally. The gobo image is formed by adding one layer then adding another layer. The multiple layers together effectively become the complex image.

Gobo selection occurs from a gobo catalog arranged in a tree structure as shown in FIG. **13**. Operation follows a logical path within the catalog. A route shown as **1300** begins the operation of selecting from a tree-like operation.

A first branch of the tree includes commonly used gobos. The gobos are also arranged by some aspect of their look, including categories shown as geometric, random, pictures. Within the geometric gobos, the gobos can be arranged by the class of the geometry, shown here as random, round, triangles, and the like. Again within each branch, there can be possibilities. Within the "round" selection is single circle, two circles, three circles, four circles that are close, etc. Each route can have twenty branches.

The menu shown in FIG. 13 pops up each time a category is selected. An important feature keeps the commonly-used parts close to the root of the catalog. This can be done either by selecting those gobos which are most common and putting these in the commonly-used paths or by taking a statistical selection of those gobos which are used. For example, a memory location could store the number of times a gobo is used, and those with for example the sixty highest numbers could be stored in the commonly used paths. Preferably those stored in the commonly used paths are also selectable via their parameters. The gobo catalog arranges the gobos as a tree that is logically connected. Many gobos have multiple characteristics. Those gobos are then categorized based on all of those multiple characteristics, and those gobos can be accessed through any of the paths for any of the categories.

Each of the layers defines an action to be taken on the image that forms a stencil for the light beam projection. The layers are combined two at a time to form a composite image. Then, that composite image is combined with the next layer to form a new composite image, and so on.

The combination can be defined as any of the following:

1) A logical "and" of bits

2) A logical "or" of bits

3) A mathematical addition of the images

4) A multiplication of the images

5) A highest text precedence combination, where the brightest parts of the images are taken.

6) An exclusive or operation.

The layered output model is controlled by names. The following represents the terminology used in this embodiment.

Gobo

The object from the catalog with no associated effects.

Effect

Tools and filters that modify a gobo.

Layer

A layer may have one of the following:

Gobo

Gobo with single or multiple effects

Effect

A combination of a gobo and effects within a layer only modifies effects the gobo within that layer.

Four layers of this type can form a Composite Image.

Layers 1–4 store in the gobo palette or directly into cues as "orphans".

Effect Layer

An Effect Layer is a special type of layer that does not contain a gobo. This layer defines and effect and modifies any layers prior to the effects layer.

Component

A single gobo or effect within a layer.

Manual Layer

All functions that simulate motorized lamp operate at this layer. These effect proportionally, or completely, anything that is stored in the image palette. The output of this layer is stored into cues as individual cue records.

Simple and Complex images do not reside on the manual layer.

Image

The combination of layers 1–4, using any of the combination techniques described above.

Simple Image

A one layer image.

Complex Image

An image having more than one layer.

Composite Image

The combination of images with the manual layer.

The image layers described herein facilitate constructing and editing an image as described herein.

The gobo menu is shown in FIG. 13. A paper based gobo catalog exists in a similar form. The gobo menu of FIG. 13 is displayed on the Menu panel. The gobos are organized by type and also categorically.

Each gobo is assigned a catalog number. This number can be used as a quick way to select a gobo. It is possible at any time to enter the numeric catalog number to access a gobo without entering the gobo menu.

Gobos selected from the Menu panel are sent to the selected fixtures and can then be stored into cues or into a palette.

A user-definable portion of the catalog is used for custom images.

Variable/Animation Gobos

A variable gobo is selected from the Menu panel. The appropriate number of unassigned or "wild" encoders automatically load. These encoders obtain control over the variable. A variable gobo can be changed in real time control via the wild encoders.

The effects menu is an additional menu mode that contains a list of all filters or tools used to modify an image. This does NOT include motorized functions.

The effects menu gives the user access to effects that may not be loaded with control on an MPD or a wild encoder.

Auto/Manual Palette Color

Any palette index may be prescribed a key color for organizational purposes.

The layer editor is an expanded palette editor that allows views and control while building a complex image.

This editor may be used or accessed manually, from a palette, or from within an Image Cue Record.

The editor includes copy/paste/delete options within the editor for layer modification.

The Menu panel is used to display layer contents.

The Manual Layer

Most components of the manual layer are controlled from a manual panel driver, Wild Encoder, or Numeric entry. Color, focus, and shutters are selected, on the manual layer from palettes.

Components of the manual layer can store into cues as individual cue records.

Hardware limitations may determine how many effects can be simultaneously processed. Until hardware is tested, a reasonable number of effects may be 12.

Filters

A new filter is added to aid effect management, called "Effect".

Image Records

An Image Record contains all the information for a simple or complex image.

Timing And Delay defines

Timing on components.

Timing on layers.

Timing on Image Records.

Timing on manual panel drivers.

Targeting Cues and Chases

Targeting

A cue record that adds or changes an effect in layer(s) 1–4 without changing the stored value of the Image.

Verbal explanation.

Combo Palette

Layer Maps

Layer Palette

Layer Presets

Operating Modes

Shutter

Fade

Although only a few embodiments have been described in detail above, those having ordinary skill in the art certainly understand that modifications are possible.

What is claimed is:

1. A method of shaping a beam of light, comprising:

providing a device which shapes a stage lighting beam based on a digital electronic signal that is applied thereto;

driving said device using a plurality of gobo layers forming said digital electronic signal, at least a first of said layers being an electronic image of a gobo to shape the beam and a second layer forming an electronic representation of a filter for the gobo image produced

by said first layer, said second layer operating to change the image of the first layer.

2. A method as in claim 1, wherein said filter causes scaling of the gobo image.

3. A method as in claim 1, wherein said filter is a decay of the gobo image.

4. A method as in claim 1, wherein said filter is a blur of the gobo image which causes the image to be out of focus in some locations.

5. A method as in claim 1, wherein said filter is a simulation of the effect of the optical glass in an analog gobo.

6. A method as in claim 1, wherein said filter includes information which is arithmetically combined with the first image.

7. A method as in claim 6, wherein said arithmetic combination is via one of an "and" and "or", or an "exclusive or".

8. A method of representing a plurality of different gobos, comprising:

forming a manual, having different parts which represent said different gobos;

adding filters for said gobos to said manual, said filters representing effects which can be carried out on other ones of said gobos; and

allowing selection of any of said gobos and/or any of said filters over a common user interface.

9. A method as in claim 8, wherein said filter is a filter that one of scales the image, decays the image, or blurs the image.

10. A method of defining an image to be projected by a light beam projector, comprising:

defining a desired gobo to be displayed as a multilayered image, where at least one of a plurality of layers includes electronic information representing at least one gobo, and at least one other layer has electronic information with the capability of including at least one effect for the at least one gobo.

11. A method as in claim 10, wherein the at least one layer includes a gobo defined by an image file, and all other layers operate on the first layer by modifying said image file.

12. A method as in claim 11, wherein a second layer includes a filter that operates on a gobo in a first layer.

13. A method as in claim 11, wherein an operation of a second layer is applied to an entire image, formed of all the layers.

14. A method as in claim 11, wherein an operation of a second layer is applied to only specified layers.

15. A method as in claim 14, wherein said operation is applied incrementally to the entire image.

16. A method as in claim 11, further comprising

forming a manual, having different parts which represent a plurality of said gobos;

adding filters for said gobos to said manual, said filters representing effects which can be carried out on other ones of said gobos; and

allowing selection of one of said gobos and/or filters over a common user interface.

17. A method as in claim 16, wherein said manual is defined as trees with branches.

18. A method as in claim 17, wherein said branches are defined according to both commonality of use and by categories.

19. A method of arranging gobos in a catalog, comprising;

organizing the gobos both in terms of both commonality of use, and category of their actual formation in a tree structure, which has branches at different locations; and

keeping more common gobos closer to a bottom portion of the tree.

20. A method as in claim 19, wherein said organizing comprises organizing into a manual, having different parts which represent said gobos;

adding filters for said gobos to said manual, said filters representing effects which can be carried out on other ones of said gobos; and

allowing selection of one of said gobos and/or filters over a common user interface.

21. A method as in claim 19, wherein at least one of the gobos is listed multiple times in the catalog.

22. A method as in claim 21, further comprising using items from the catalog to form a plurality of layers, and combining said layers to form a composite image.

23. A method as in claim 22, wherein said composite image is formed by a mathematical combination of bits in the catalog.

24. A method as in claim 23, wherein said combination is a logical AND, a logical OR, a mathematical addition or multiplication, or a highest text precedent combination where brightest parts of the image are taken, or an exclusive OR.

25. A method as in claim 19, wherein said gobo is a variable animation gobo.

26. A method as in claim 25, wherein said variable animation gobo includes a menu mode.

* * * * *

TO: Commissioner Of Patents P.O. Box 1450 Alexandria, VA 22313 (571) 272-8800	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A *PATENT* OR *TRADEMARK*

In compliance with the provisions of 35 § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Southern District on the following ☑ Patents or ☐ Trademarks:

DOCKET NO.	DATE FILED	
08 CV 6331	7/15/08	United States District Court Southern District of New York 300 Quarropas Street White Plains, NY 10601-4150

PLAINTIFF	DEFENDANT
Production Resource Group, LLC	SGM Technology for lighting SpA, et al

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1. 6,711,411	8/3/04	PRG
2. 6,891,656	5/10/05	PRG
3. 6,919,916	7/19/05	PRG
4. 7,161,562	1/9/07	PRG
5.		

In the above-entitled case, the following patent(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	**DATE OF PATENT OR TRADEMARK**	**HOLDER OF PATENT OR TRADEMARK**
1.		
2.		
3.		
4.		
5.		

In the above-entitled case, the following decision has been rendered or judgment issued:

DECISION/JUDGMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1 Upon *initiation* of action, mail this copy to Commissioner
Copy 2 Upon *filing document adding patent(s)*, mail this copy to Commissioner
Copy 3 Upon *termination of action*, mail this copy to Commissioner
Copy 4 Case file Copy